NUMERICAL GRID GENERATION AND POTENTIAL AIRFOIL ANALYSIS AND DESIGN

APPROVED BY SUPERVISORY COMMITTEE:

G. F. Carey

D S Dawning

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE *(When Data Entered)*

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER AFIT/CI/NR 88-72	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
4. TITLE *(and Subtitle)* NUMERICAL GRID GENERATION AND POTENTIAL AIRFOIL ANALYSIS AND DESIGN		5. TYPE OF REPORT & PERIOD COVERED MS THESIS 6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) MATO F. SILADIC		8. CONTRACT OR GRANT NUMBER(s)
9. PERFORMING ORGANIZATION NAME AND ADDRESS AFIT STUDENT AT: UNIVERSITY OF TEXAS AT AUSTIN		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
11. CONTROLLING OFFICE NAME AND ADDRESS		12. REPORT DATE 1988 13. NUMBER OF PAGES 131
14. MONITORING AGENCY NAME & ADDRESS *(if different from Controlling Office)* AFIT/NR Wright-Patterson AFB OH 45433-6583		15. SECURITY CLASS. *(of this report)* UNCLASSIFIED 15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT *(of this Report)*

DISTRIBUTED UNLIMITED: APPROVED FOR PUBLIC RELEASE

17. DISTRIBUTION STATEMENT *(of the abstract entered in Block 20, if different from Report)*

SAME AS REPORT

18. SUPPLEMENTARY NOTES

Approved for Public Release: IAW AFR 190-1
LYNN E. WOLAVER Lynn Wolaver 19 July 88
Dean for Research and Professional Development
Air Force Institute of Technology
Wright-Patterson AFB OH 45433-6583

19. KEY WORDS *(Continue on reverse side if necessary and identify by block number)*

20. ABSTRACT *(Continue on reverse side if necessary and identify by block number)*

ATTACHED

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE *(When Data Entered)*

Part I
ELLIPTIC AND PARABOLIC PARTIAL DIFFERENTIAL EQUATIONS IN NUMERICAL GRID GENERATION

1 INTRODUCTION

The primary objective of this study is to explore the feasibility of using parabolic partial differential equation techniques for numerical grid generation for two-dimensional aerodynamic configurations. The contents include a discussion of grid generation concepts and schemes in the literature, iterative methods for numerical grid generation and two differential grid generation schemes: (1) *an elliptic*, and (2) *a parabolic scheme*. A detailed mathematical and numerical representation of both schemes is given. The main purpose in the treatment of the elliptic scheme is to introduce the necessary transformations to provide equations that will establish the particular parabolic scheme development.

The grid generation scheme is derived from a pair of model parabolic

equations in which the coefficients are determined by use of a related set of two- dimensional elliptic partial differential equations. Coefficients that have effects similar to the exponential forcing functions for grid spacing control in elliptic grid generation are determined. This formulation involves discretization of the governing equations on a quasi-uniform grid in the computational plane and using linear interpolation between the current grid line and outer boundary.

The basic idea in grid generation by use of parabolic PDE's is to specify the body surface as an "initial" boundary, and treat the outer boundary as a constraint. Then coordinate lines are generated by marching outward from the body surface.

The parabolic grid generation method is extended to multicomponent airfoils, arbitrary multiple bodies and cascade grid generation for the first time. We present representative results as a measure of the efficiency of the parabolic scheme.

2 GRID GENERATION

Numerical grid generation has become an integral part of computational fluid dynamics (CFD) and is one of the most important topics in the development of flow solutions in complex flow domains. The grid here is understood to be an organized set of points formed by the intersections of the coordinate lines of a numerically generated boundary-conforming curvilinear coordinate system having the same dimensionality as the physical region. The main feature of such a system is that some coordinate line (surface in three dimensions) is coincident with each segment of the physical boundary. There are basically two decision stages involved in the

To my wife and my children

NUMERICAL GRID GENERATION AND POTENTIAL AIRFOIL ANALYSIS AND DESIGN

by

MATO F. SILADIC, B.SC.

THESIS

Presented to the Faculty of the Graduate School of
The University of Texas at Austin
in Partial Fulfillment
of the Degree of

MASTER OF SCIENCE IN ENGINEERING

THE UNIVERSITY OF TEXAS AT AUSTIN
May, 1987

ACKNOWLEDGEMENTS

The author would like to express his sincere gratitude to Dr. Graham F. Carey for invaluable advice and encouragement during the preparation of the thesis. Special thanks are expressed also to Dr. David S. Dolling for serving as member of the author's supervisory committee.

I would also like to thank my associates in the CFD Lab, particularly Mr. Chung Y. Huang, Mr. Stephen Kennon and Mr. Steve Bova and the faculty and stuff of the Department of Aerospace Engineering an Engineering Mechanics. Finally, I would like to thank my wife, Ankica, for understanding, moral support, sleepless nights and lost weekends.

This research was supported by the Yugoslav Government and U.S. Air Force, to whom I am greatly indebted.

TABLE OF CONTENTS

ACKNOWLEDGEMENTS IV

TABLE OF CONTENTS V

I ELLIPTIC AND PARABOLIC PARTIAL DIFFERENTIAL EQUATIONS IN NUMERICAL GRID GENERATION 1

1 INTRODUCTION 1

2 GRID GENERATION 2

2.1 ALGEBRAIC METHODS 4

2.2 CONFORMAL MAPPING METHODS 6

2.3 DIFFERENTIAL GRID-GENERATION SCHEMES 8

2.3.1 GENERAL PROPERTIES OF SECOND ORDER PARTIAL DIFFERENTIAL EQUATIONS 8

2.3.2 ITERATIVE METHODS FOR SOLUTION OF PDE'S . . . 10

2.4 ELLIPTIC GRID GENERATION SCHEMES 14

2.5 HYPERBOLIC SCHEMES 16

2.6 PARABOLIC SCHEME 17

2.7 ADAPTIVE GRIDS 18

3 NEED FOR BOUNDARY-FITTED COORDINATE SYSTEM 19

3.1 TYPE OF GRIDS 21
3.2 TRANSFORMATIONS BETWEEN PHYSICAL AND COMPUTATIONAL PLANE 22

4 AN ELLIPTIC SCHEME AND APPLICATION 24
4.1 COORDINATE SYSTEM CONTROL 29
4.2 APPLICATION OF THE ELLIPTIC SCHEME 30

5 PARABOLIC SCHEME AND APPLICATION 33
5.1 GENERAL OBSERVATIONS 33
5.2 NUMERICAL DEVELOPMENT OF PARABOLIC SCHEME 35
5.3 GRID SPACING CONTROL 39
5.4 EFFECT OF THE COEFFICIENTS ON GRID GENERATION 40
5.5 PRACTICAL APPLICATION AND RESULTS 41
5.6 COMPUTATIONAL EFFICIENCY 47
5.7 LIMITATIONS OF PARABOLIC SCHEME 47
5.8 CLUSTERING OF POINTS IN GIVEN REGIONS 48

6 CONCLUSIONS 48

II POTENTIAL AIRFOIL ANALYSIS AND DESIGN 50

7 INTRODUCTION 50
7.1 REVIEW OF POTENTIAL METHODS 51
7.2 GENERAL CONSIDERATIONS 53

8 MATHEMATICAL DEVELOPMENT OF PANEL METHODS 55
8.1 POTENTIAL BASED PANEL METHOD 58

9 STREAM FUNCTION APPROACH 61
9.1 A DESIGN METHOD 65

10 DISCUSSION OF RESULTS 68
10.1 AIRFOIL ANALYSIS 68
10.2 AIRFOIL DESIGN 70
10.2.1 SINGLE AIRFOIL DESIGN 70
10.2.2 MULTICOMPONENT AIRFOIL DESIGN 71

11 CONCLUSION 73

APPENDICES

12 VORTEX PANEL METHOD WITH LINEAR VORTEX STRENGTH 74

13 CURVED PANEL DISCRETIZATION 77

BIBLIOGRAPHY 124

VITA 131

Part I
ELLIPTIC AND PARABOLIC PARTIAL DIFFERENTIAL EQUATIONS IN NUMERICAL GRID GENERATION

1 INTRODUCTION

The primary objective of this study is to explore the feasibility of using parabolic partial differential equation techniques for numerical grid generation for two-dimensional aerodynamic configurations. The contents include a discussion of grid generation concepts and schemes in the literature, iterative methods for numerical grid generation and two differential grid generation schemes: (1) *an elliptic*, and (2) *a parabolic scheme*. A detailed mathematical and numerical representation of both schemes is given. The main purpose in the treatment of the elliptic scheme is to introduce the necessary transformations to provide equations that will establish the particular parabolic scheme development.

The grid generation scheme is derived from a pair of model parabolic

equations in which the coefficients are determined by use of a related set of two- dimensional elliptic partial differential equations. Coefficients that have effects similar to the exponential forcing functions for grid spacing control in elliptic grid generation are determined. This formulation involves discretization of the governing equations on a quasi-uniform grid in the computational plane and using linear interpolation between the current grid line and outer boundary.

The basic idea in grid generation by use of parabolic PDE's is to specify the body surface as an "initial" boundary, and treat the outer boundary as a constraint. Then coordinate lines are generated by marching outward from the body surface.

The parabolic grid generation method is extended to multicomponent airfoils, arbitrary multiple bodies and cascade grid generation for the first time. We present representative results as a measure of the efficiency of the parabolic scheme.

2 GRID GENERATION

Numerical grid generation has become an integral part of computational fluid dynamics (CFD) and is one of the most important topics in the development of flow solutions in complex flow domains. The grid here is understood to be an organized set of points formed by the intersections of the coordinate lines of a numerically generated boundary-conforming curvilinear coordinate system having the same dimensionality as the physical region. The main feature of such a system is that some coordinate line (surface in three dimensions) is coincident with each segment of the physical boundary. There are basically two decision stages involved in the

discretization of a flow field. The first involves a decision about the grid-generation concept to be used, and the second involves a decision about the grid generation scheme to be employed. Several grid-generation concepts and schemes are summarized in fig 1.

Most discretization methods can be classified as based upon one or more of the concepts in fig. 1. These include single-module, multi-block, interfering, component-adaptive overlapping, and component adaptive interfacing approaches. Probably the easiest and most popular concept is the single module, in which the discretized flow domain is transformed into a single computational rectangle (in 2-D) or cube (in 3-D).

The multi-block approach corresponds to linking together several such blocks. Two methods are discussed by Rubbert and Lee [1]. The first method is to generate the grid separately within rectangular subdomains (blocks). Some of the block boundary surfaces no longer correspond to boundary surfaces of the original problem, but instead separate adjacent blocks. These are called field boundaries. Solution of the grid generation equations requires grid boundary conditions on these field boundaries. Grids generated in this manner are termed "patched grids" or "patched coordinate systems". The second method is to solve the grid generation equations in the entire block- structured domain as a single grid generation problem. In this case grids will be analytical across field faces, but solution of the problem is more complex. Grids generated in this manner are called "directly solved multi- block grids". An example of an effective multi-block approach is given by Thomas [2] for discretizing a wing fuselage configuration.

Once the desired grid-generation concept has been determined for a given flow problem, the next step is to determine the actual location of

the grid points. To do this, a variety of grid generation schemes are available. Basically, the grid-generation schemes for curvilinear grids are of two general types (fig. 2) :

1. construction by algebraic interpolation

2. numerical solution of partial differential equations

In the latter, partial differential equation systems of elliptic, parabolic or hyperbolic type may be used. Elliptic systems may be used to generate both conformal and quasiconformal mappings, the former being orthogonal. In this study, emphasis is on differential grid-generation schemes. However, for completeness we first describe algebraic methods.

2.1 ALGEBRAIC METHODS

The primary advantages of algebraic grid generation methods are that they allow explicit control of the physical grid shape and spacing and are inexpensive. The fundamental idea on which most algebraic methods are based is the use of mathematical interpolation functions to interpolate between some known (or pre-assigned) grid points (usually on the boundaries), in order to generate the grid in between these points. The interpolation method or technique may vary with different algebraic methods, but the essential idea remains the same. The points between which the interpolation is carried out need not be boundary points; they can be any specified points in the interior of the grid, through which we desire that the grid lines pass. The interpolation functions contain coefficients which are determined so that the functional values match the coordinate values at these specified points. The most difficult aspect of algebraic grid generation is the determination of functions which control a grid.

For unidirectional interpolation (interpolation in one coordinate direction alone) the interpolation methods commonly used are polynomial interpolation and spline fitting. In polynomial interpolation a single polynomial function is used, to match the coordinate values at all the specified points (e.g., Langrange interpolation). In some polynomial interpolation methods it is also possible to specify the slope of the function at any or all of the specified grid points (e.g., Hermite interpolation). The disadvantage of polynomial interpolation is that, as the number of specified quantities (i.e., points and/or slope at these points) increase, the order of the polynomial increases. This may lead to undesirable oscillations in the behavior of the polynomial in between the specified grid points. This problem can be avoided if spline fitting is used instead of polynomials. In spline fitting, instead of using a single polynomial for all the points, low order piecewise polynomials are used to interpolate the specified grid points. Slope continuity is enforced at each point so that smooth grid lines are obtained. The advantage of using splines is that grid lines can be generated relatively easily and oscillation of the grid lines is reduced. An effective procedure based on the description of two exterior boundaries and the application of either linear or Hermite cubic polynomial interpolation to compute the interior grid is given in Smith [3]. For cubic interpolation surface derivatives combined with magnitude coefficients control the orthogonality of the grid at and near the boundaries.

For example, non-polynomial interpolation functions are sometimes used when the variation in spacing between the specified grid points is large. The functions commonly used for this are the exponential functions and the hyperbolic sine and tangent functions. In the multisurface method, developed by Eiseman [4], interpolation is carried out between an inner and

outer boundary, with a series of control surfaces in between. This involves the use of field vectors tangent to the coordinate curve passing across these surfaces. The most common form is transfinite interpolation, which is essentially an interpolation between curves or surfaces rather than points. It is called "transfinite" because it matches coordinate values on an entire curve or surface. For example, Erikson [5] applies transfinite interpolation for three -dimensional grid generation about wing - body configurations. Some examples for airfoils and cascade configurations are given by Gordon and Thiel [6] and Eiseman [4, 7].

2.2 CONFORMAL MAPPING METHODS

In conformal mapping methods analytic functions are used for transforming the physical domain into intermediate domains wherein the grid generation problem is simpler, and subsequently for remapping the grid into the physical or computational domain. These methods usually lead to reasonable forms for the transformed partial differential equations. In addition they can be used for generating orthogonal grids. A disadvantage of conformal methods is that they provide little control over the grid point distribution for general domains.

The general procedure for grid generation using conformal methods consists of two steps. The first step consists of determining an appropriate mapping, or sequence of mappings, that would transform a given physical domain into a simple region. The second step consists of generating the orthogonal grid in the computational domain. This grid is then remapped to obtain an orthogonal grid in the physical domain.

Contours such as single airfoils are generally mapped to "near-circles" by one or more simple transformations, and then the "near-circle" is mapped

to a circle by a series transformation, (i.e., the Theodorsen procedure (Akai and Mueller [8])). It is necessary for convergence that "the near-circle" be sufficiently close to circular. Also, there are some very interesting ideas for grid generation about multi-component airfoils - Halsey [9,10] discusses three methods. The first method treats multicomponent airfoils by a sequence of transformations which map each airfoil to a circle in succession, while maintaining previously established circles. The second approach involves mapping each airfoil to a circle with no special consideration of the others. This process generally requires only a few iterations to converge. The third approach involves connecting all of the bodies in a string and mapping the resulting (effective) single body. This procedure is the simplest, but will not give satisfactory grids for closely spaced bodies in general. Similar ideas are presented by Harrington [11] where the bodies are all mapped to rectangles and by Ives [12] where a Karman-Trefftz transformation is used to map airfoils to the near circles.

Although the complex variable techniques by which conformal transformations are usually generated are inherently two-dimensional, certain more general cases can be treated by rotating or stacking two-dimensional systems. For example, Dulikravich [13] has developed multilevel three-dimensional, C-type periodic, boundary conforming grids for the calculation of realistic turbomachinery and propeller flow fields. The method is based on two analytic functions that conformally map a cascade of semi-infinite slits to a cascade of doubly infinite strips on different Riemann sheets. For the same type of problems Anderson, Davis, Hankins and Edwards [14] apply Shwarz- Christoffel transformation and demonstrate the validity of the grid in calculations of viscous turbulent flow through a turbine inlet duct.

2.3 DIFFERENTIAL GRID-GENERATION SCHEMES

Differential grid-generation schemes have received widespread attention because of their versatility and the ease with which they can be applied. Here the essential idea is to generate the coordinate mapping (and hence the grid) as the discrete approximate solution to a partial differential equation, matching the given boundary shape as appropriate boundary or initial data. The underlying characteristics of differential grid generation schemes are depicted in fig. 3. We first recall some fundamentals for second order partial differential equations and their solutions.

2.3.1 GENERAL PROPERTIES OF SECOND ORDER PARTIAL DIFFERENTIAL EQUATIONS

The classification scheme for second order PDE's depends on the nature of their characteristics. In the case of two independent variables, characteristics are lines in the plane of the independent variables along which "signals" can propagate.

Consider the second order quasilinear PDE in two independent variables of the form

$$A\Phi_{xx} + B\Phi_{xy} + C\Phi_{yy} = 0 \tag{1}$$

where A, B, C and D may all be functions of x, y, Φ, Φ_x and Φ_y. The classification of this equation depends on the sign of $B^2 - 4AC$. If $B^2 - 4AC > 0$ the equation is called hyperbolic; if $B^2 - 4AC = 0$, the equation is parabolic; if $B^2 - 4AC < 0$, the equation is elliptic.

- HYPERBOLIC PDE'S

Hyperbolic PDE's posses two families of real characteristics. Physical systems that are governed by hyperbolic equations involve signals that propagate at finite speed. They are frequently posed in domains that extend to infinity in the timelike coordinate and are thus unbounded in this direction. The spatial coordinate may or may not be bounded. In either case one typically specifies two initial conditions at $t = 0$. The characteristics define the range of influence of initial data. If the spatial region is bounded, boundary conditions are also specified; otherwise, we have a pure initial-value problem. The prototype (model) hyperbolic equation is the wave equation

$$\Phi_{tt} - c^2\Phi_{xx} = 0 \tag{2}$$

where c is the wave speed.

- PARABOLIC PDE'S

Parabolic partial differential equations can be regarded as the limit of hyperbolic equations in which the propagation speed of the signal becomes infinite. The prototype (model) parabolic equation is the heat equation

$$\Phi_t = D\Phi_{xx} \tag{3}$$

Since equation (3) is first order in time, only one initial condition at $t = 0$ is necessary, and there is only one set of characteristics. The domain of solution is unbounded in time and the spatial domain may be unbounded or finite.

- ELLIPTIC PDE'S

Elliptic partial differential equations have no real characteristics. In elliptic problems every point in the solution domain is effected by disturbances at every other point. The prototype (model) elliptic PDE is the

Laplace's equation

$$\Phi_{xx} + \Phi_{yy} = 0 \tag{4}$$

Boundary conditions are provided by giving the value of the dependent variable, its normal derivative, or some linear combination of the two, at each point on the boundary.

2.3.2 ITERATIVE METHODS FOR SOLUTION OF PDE'S

Iterative solutions and time stepping techniques to steady state are closely related. We consider iterative grid generation for elliptic systems and marching algorithms for parabolic or hyperbolic systems.

An iterative method applied to model PDE (4) is a procedure of the type

$$[A_1]\Phi^{k+1} = [A_2]\Phi^k + d \tag{5}$$

where A_1 and A_2 are matrices, d is a vector and k indicates the iteration level. Given some initial guess $\Phi^{(0)}$ for the solution , we use (5) to find $\Phi^{(1)}$ and $\Phi^{(2)}$ and so on. If this is to be a satisfactory means of solving the original equation, it should have the following properties:

1. it should converge to the exact solution of the original equation. That is we should have $\lim_{k\to\infty} \Phi^{(k)} = \Phi$, where Φ is a solution of $[A]\Phi = b$

2. the convergence should be rapid if the method is to be efficient; that is the above limit should be reached quickly.

3. so that each step does not require much computation, the matrix A_1 should be easy to invert and the matrix A_2 should be as simple as possible to facilitate the computation of $[A]_2\Phi^k$.

In this section we consider Jacobi, Gauss-Seidel, SOR and ADI iterative methods

- JACOBI METHOD

In the Jacobi method each new value of a function is computed entirely from old values. Applied to model PDE equation (4) with uniform spacing $\Delta x = \Delta y = 1$ the method can be written in component form as

$$\Phi_{i,j}^{k+1} = 0.25(\Phi_{i+1,j}^{k} + \Phi_{i-1,j}^{k} + \Phi_{i,j+1}^{k} + \Phi_{i,j-1}^{k}) - 0.25b_{i,j} \qquad (6)$$

where i - represents row, j - column and k - iteration level. New values of $\Phi_{i,j}$ are computed by averaging the values of neighbors at the preceding iteration and adding the inhomogeneous (boundary condition) term.

- GAUSS-SEIDEL METHOD

When we compute $\Phi_{i,j}$ in a Jacobi method, we have already calculated the values of $\Phi_{i-1,j}$ and $\Phi_{i+1,j}$. Since we expect at each iteration level the new values to be better approximated than the old ones, improvement could be achieved by using updated values. This leads, for model PDE eq. (4), to

$$\Phi_{i,j}^{k+1} = 0.25(\Phi_{i-1,j}^{k+1} + \Phi_{i+1,j}^{k} + \Phi_{i,j-1}^{k+1} + \Phi_{i,j+1}^{k}) - 0.25b_{i,j} \qquad (7)$$

which is the basis of the Gauss-Seidel method. Sufficient condition for convergence of the Gauss-Seidel method is diagonal-dominance of [A].

- SUCESSIVE OVER-RELAXATION (SOR)

Instead of using $\Phi_{i,j}^{k+1}$ as the input for the next iteration, we can extrapolate the preceding results. This leads to the procedure known as extrapolation

or over-relaxation. To introduce this idea, suppose that we use Gauss-Seidel method to compute

$$\tilde{\Phi}_{i,j}^{k+1} = [M_j]\Phi_{i,j}^{k} + d_{i,j} \tag{8}$$

Using the key idea of extrapolation instead of using $\Phi_{i,j}^{k+1}$ as the new iterate we modify it by weighted averaging as

$$\Phi_{i,j}^{k+1} = \omega\tilde{\Phi}_{i,j}^{k+1} + (1-\omega)\Phi_{i,j}^{k} \tag{9}$$

here, $\tilde{\Phi}_{i,j}^{k+1}$ is the most recent value of $\Phi_{i,j}$ calculated from (8), $\Phi_{i,j}^{k}$ is the value from previous iteration and $\Phi_{i,j}^{k+1}$ is the newly adjusted value of $\Phi_{i,j}$. For eq. (9) to be an extrapolation and stable, overrelaxation factor ω should be such that $1 < \omega < 2$. In some problems (occasionally for nonlinear problems) underrelaxation $0 < \omega < 1$ is employed.

- ADI METHOD

The basic idea of ADI method is to introduce intermediate iteration level $k + 1/2$ with operator splitting. The method can be presented as follows: Let us split matrix $[A]$ into two matrices $[A] = [H] + [V]$ such that

1. $[H] + r[I]$ and $[V] + r[I]$ are nonsingular for any $r > 0$.

2. It is convenient to solve

$$([H] + r[I])u_1 = b_1 \tag{10a}$$

$$([V] + r[I])u_2 = b_2 \tag{10b}$$

where b_1 and b_2 are any vectors and $r > 0$. Here the term "convenient" is taken to mean 'easy to solve on a computer' and thus we shall select $[H]$ and $[V]$ as tridiagonal matrices (or matrices convertible to tridiagonal

form). In the Peacman-Rachford method (Lapidus and Pinder [15]), we select a value $r > 0$ and write

$$([H] + r[I])u^{k+1/2} = b - ([V] - r[I])u^k \tag{11}$$

$$([V] + r[I])u^{k+1} = b - ([H] - r[I])u^{k+1/2} \tag{12}$$

with the vector $u^{k+1/2}$ being an intermediate vector. Thus (11) is used with $u^{(k)}$, the matrices $[H]$ and $[V]$, and r to calculate $u^{(k+1/2)}$; then (12) is used with r to evaluate $u^{(k+1)}$. In this way the iteration proceeds from $u^{(0)}$, $u^{(1)}$... . The idea here is the selection of $[H]$ and $[V]$ so that (11) uses lines in the x direction only and (12) uses lines in the y direction only. In order to consider convergence of the method let us define error vector $\epsilon^{(k)} = u^{(k)} - u$, where u, the correct solution, satisfies

$$([H] + [V])u = b \tag{13}$$

It then follows that

$$\epsilon^{(k+1)} = [G]_r \epsilon^{(k)}$$

where $[G]_r = ([V]+r[I])^{-1}([H]-r[I])([H]+r[I])^{-1}([V]-r[I])$ is the iteration matrix. It is necessary for convergence that the spectral radius $[G]_r$ satisfies $r([G]_r) < 1$. If matrices $[H]$ and $[V]$ are positive define, the above condition holds. For model equation (4) the method can be written (with $r = 1$) as

- first step:

$$\phi_{i-1,j}^{k+1/2} - 2\phi_{i,j}^{k+1/2} + \phi_{i+1,j}^{k+1/2} = -(\phi_{i-1,j}^{k} - 2\phi_{i,j}^{k} + \phi_{i+1,j}^{k}) \tag{14}$$

- second step:

$$\phi_{i-1,j}^{k+1} - 2\phi_{i,j}^{k+1} + \phi_{i+1,j}^{k+1} = -(\phi_{i-1,j}^{k+1/2} - 2\phi_{i,j}^{k+1/2} + \phi_{i+1,j}^{k+1/2}) \tag{15}$$

The iteration begin with an initial estimate of the $\Phi^{(0)}$ vector, and computation is terminated when successive calculations agree within a given tolerance.

Except for the additional work of recalculating right sides, this method has little computational effort and converges fast. The coefficients matrices are always the same, so the reduction step need be performed only once.

2.4 ELLIPTIC GRID GENERATION SCHEMES

Elliptic schemes require the specification of data on the entire boundary of the domain. The location of the interior grid points is then determined by solving a set of elliptic partial differential equations. The most commonly used are the Laplace and Poisson equations. In the former, with equispread boundary points, the coordinate lines will tend to be equally spaced in the absence of boundary curvature, because of the smoothing effect of the Laplacian. In the Poisson case mesh control (coordinate lines clustering and inclination) can be accomplished by means of an appropriate choice of forcing functions.

The Poisson system defines the computational (ξ, η) coordinates by

$$\xi_{xx} + \xi_{yy} = P(\xi, \eta) \tag{16a}$$

$$\eta_{xx} + \eta_{yy} = Q(\xi, \eta) \tag{16b}$$

where $P(\xi, \eta)$ and $Q(\xi, \eta)$ represent forcing functions that are specified and depend on the nature of the problem and the desired grid point distribution.

The primary advantages of such elliptic systems for grid generation are: *(1) they frequently generate one-to-one mapping; (2) the coordinate lines are smooth owing to the inherent smoothness in the solution of elliptic*

systems; and (3) boundary slope discontinuities are not propagated into the field.

Grid generation schemes formulated using the Laplace and Poisson equations have been extensively applied for various two- and three-dimensional configurations. For example, Thompson, Thames and Mastin [16] and Holst [17] control the interior grid distribution using exponential forcing functions which contain adjustable parameters. The selection of these parameters, however, depends on the shape of the body surface. They provide the means to move the coordinate lines around but not the proper amplitudes and decay factors necessary to achieve desired spacing distributions. Hence, this technique requires interaction from the user.

Sorenson and Steger [18] introduce automatic mesh-point clustering near the boundary and Sorenson [19], applies the idea to elliptic grid generation for discretizing the flow about the augmentor- wing using a component adaptive grid interfacing technique. His method differs from that of Thompson, Thames, and Mastin [16] in that it uses forcing terms $P(\xi, \eta)$ and $Q(\xi, \eta)$, which yield the ability to arbitrarily impose two types of control on the grid at boundaries. The first type of control is of the angle on inclination at which the lines of constant ξ intersect the boundaries; the second type of control is on the distance between the airfoil and lines $\eta = constant$.

Coleman [20] applies the Laplace or Poisson equations to produce a mesh which adapts to the physical region of interest. The method provides arbitrary segmentation in the physical plane and mapping to a union of rectangular grid sections.

Chen and Obasih [21] introduce contracting functions for the grid inside the computational domain. They treat contraction functions $P(\xi, \eta)$ and

$Q(\xi,\eta)$ in the original Poisson equation as unknowns, and require orthogonality of coordinate lines near boundaries, to determine expressions for $P(\xi,\eta)$ and $Q(\xi,\eta)$. Similar ideas are given by Visbal and Knight [22] and Shieh [23].

2.5 HYPERBOLIC SCHEMES

Hyperbolic schemes are obtained by integrating a hyperbolic PDE for the grid lines and require Cauchy data on the boundary. The grid may be generated explicitly by marching outward from the initial boundary. For example Thompson, Warsi and Mastin [16] discuss the system

$$x_\xi y_\eta - y_\xi x_\eta = V(\xi,\eta) \tag{17}$$

$$x_\xi x_\eta - y_\xi y_\eta = 0 \tag{18}$$

for generating a grid. Here $V(\xi,\eta)$ is the Jacobian of the transformation and represents the area in physical space for a given unit area element in computational space. If $V(\xi,\eta)$ is given as function of position then equation (17) can be used for grid spacing control in the physical plane. Equation (18) represents a measure of the orthogonality of grid lines in the physical space. After linearization of equations (17,18) about a known state $(\overline{x},\overline{y})$ we have the system

$$[A]r_\xi + [B]r_\eta = f \tag{19}$$

where

$$r = \begin{bmatrix} x \\ y \end{bmatrix}; [A] = \begin{bmatrix} \overline{x}_\eta & \overline{y}_\eta \\ \overline{y}_\eta & -\overline{x}_\eta \end{bmatrix}; [B] = \begin{bmatrix} \overline{x}_\xi & \overline{y}_\eta \\ -\overline{y}_\xi & \overline{x}_\xi \end{bmatrix}; f = \begin{bmatrix} 0 \\ V + \overline{V} \end{bmatrix}$$

The eigenvalues of $[B]^{-1}[A]$

$$\lambda_{1,2} = \sqrt{\frac{\overline{x}_\eta^2 + \overline{y}_\eta^2}{\overline{x}_\xi^2 + \overline{y}_\xi^2}} \tag{20}$$

are real and positive which implies that system (19) is hyperbolic in the η direction and can be marched in η as long as $\overline{x}_\xi^2 + \overline{y}_\xi^2 \neq 0$

In generating a grid with this scheme first assume the body surface is the $\eta = 0$ surface and specify the distribution of points along the body. Next the quantity $V(\xi, \eta)$ in eq.(17) is required. Steger and Sorenson [25] suggest that $V(\xi, \eta)$ can be determined by laying out a straight line with length equal to that of the body surface, and then lay out the body point distribution on this line (i.e., arc length distribution of surface points). Next a line parallel to the first line is drawn on the $\eta = \textit{constant}$ surface as desired. Once this is done, the quantity $V(\xi, \eta)$ is determined by estimating the area elements of the grid. Application of a hyperbolic scheme for airfoil problems is given by Steger and Chaussee [26] and for the three-dimensional grid about a shuttle cross-section by Kutler [27].

2.6 PARABOLIC SCHEME

Grids may be generated by parabolic schemes explicitly by marching as in the hyperbolic grid-generation method from an "initial" boundary. As in the elliptic schemes diffusion smoothes out any irregularities in the grid due to the shape of the initial line (inner boundary). The *importance of the marching algorithm* is twofold: first, *the computational time required may frequently be only a very small fraction of that for elliptic grid generation*; second, *storage required during grid generation can be substantially reduced from that required by the elliptic grid generation method.* We demonstrate these points for a particular form of parabolic scheme in later numerical experiments.

The parabolic schemes are applicable to both, two- and three-dimensional problems. Nakamura [28] applies a parabolic scheme with simplified

second derivatives to O-type and H-type grids for single airfoils and Nakamura [29] applies parabolic differencing scheme to one coordinate variable of a three-dimensional elliptic equation set for fuselage-wing configurations. Edwards [30] extends this three-dimensional grid generation algorithm to two of the coordinate variables while the third is centrally differenced and generates a grid about a wing-fuselage and an aircraft configuration.

Parabolic systems have previously been considered to be limited in the configurations that they can handle. In the present study we demonstrate that parabolic schemes can be extended to complex configurations with orthogonality satisfied on inner boundaries, and apply the method to representative multi-airfoil problems.

2.7 ADAPTIVE GRIDS

Adaptive grids are dynamic grids in which the grid points are automatically readjusted as the solution evolves. Some aspect of the developing solution must be used as the error indicator for redistributing the grid points. We note that readjustment of the grid points must yield a grid of acceptable smoothness and orthogonality. The grid points should be clustered in regions of large solution variations and be well graded into such regions. As an example, Brackbill [31] employs an objective function which contains a measure of grid smoothness, orthogonality, and volume variation that is minimized over a class of admissible grids. The smoothness of the transformation is represented by the integral

$$I_S = \int_D [(\nabla\xi)^2 + (\nabla\eta)^2] dV \tag{21}$$

A measure of orthogonality is provided by

$$I_O = \int_D (\nabla\xi \nabla\eta)^2 J^3 dV \tag{22}$$

and an error indicator measure (weighted by the Jacobian)

$$I_V = \int_D W J dV \tag{23}$$

where W is a given weighting function and J is a measure of cell volume, usually the Jacobian of transformation. If I_V is minimized for a fixed $W(x, y)$, J (the grid size) is made small where W is large and vice versa. The weight function $W(x, y)$ is the error indicator and is to be a function of some measure of the solution error, so that the spacing will be reduced where the error is large. As a final objective function a weighted sum of (21) - (23) is taken. The inclusion of (21) and (22) as integral constraints in the objective function ensures that the optimal grid has reasonable smoothness and orthogonality, respectively. Brackbill [31] implements the adaptive control through terms in an elliptic generating system and Saltzman and Brackbill [32] demonstrate results for multiple shock reflections in a wind tunnel problem. Anderson and Rai [33] give a procedure based on an analogy with electrostatic charge attraction which is applicable to any coordinate system and demonstrate the effectiveness of the shock aligning scheme for a straight oblique shock in a uniform supersonic flow.

3 NEED FOR BOUNDARY-FITTED COORDINATE SYSTEM

There arises in all problems concerned with the numerical solution of partial differential equations the need for accurate numerical representation of boundary conditions. For example, in the present study we seek to define points of a finite difference grid constructed on coordinate lines that coincide with the boundary. That is one coordinate variable can be specified to

be constant on each of boundaries, and a monotonic variation of the other coordinate around each boundary can be specified as the data for our discretized partial differential equation problem. It then remains to generate values of these coordinates in the field from prescribed boundary or initial values. There must, of course, be a unique correspondence between the basic coordinate system and the curvilinear coordinates; i.e., the mapping of the physical region onto a transformed computational region must be one-to-one, so that every point in the physical field corresponds to one, and only one, point in the transformed field, and vice versa. Coordinate lines of the same family must not cross, and lines of different families must not cross more than once (i.e., the Jacobian of the transformation must be nonsingular at each point). Further, the grid lines should be smooth to provide continuous transformation derivatives. Grid points should be closely spaced in the physical domain where large numerical errors are expected and excessive grid skewness should be avoided because it can sometimes amplify truncation error.

Since the boundary-fitted coordinate system has coordinate lines coincident with the surface contours of all bodies present, all boundary conditions for the problem can be expressed at grid points without recourse to interpolation or projection, and normal derivatives on the bodies can be represented using finite differences between grid points even though the coordinate system may not be orthogonal at the boundary. The transformed equations can then be approximated using finite difference expressions and solved numerically in the transformed plane.

3.1 TYPE OF GRIDS

Requirements on the type of grid to be used in the physical domain are mainly problem dependent. Three grid types for aerodynamic flow problems are in common use and have been termed *H-type, O-type and C-type.*

- A *H-type grid* (fig. 4a) provides excellent resolution of the flow field at upstream and downstream infinity. It is also the simplest grid to generate. At the same time, H-type grids do not provide an accurate treatment of rounded leading and trailing edges and grid points in the flow domain away from the boundaries are not well clustered.

- An *O-type grid* (fig. 4b) represents a coordinate system having lines encircling a body. It gives very poor resolution at infinity, but provides very good resolution for blunt and rounded edges, and uses few grid points

- A *C-type grid* (fig. 4c) indicates a coordinate system with lines emanating from a boundary, passing around a body and returning to the boundary. The grid represents a combination of an O-type grid in the upstream region and H-type grid in the downstream region. This type of grid provides a good treatment of all boundary and periodicity conditions including wake treatment and supersonic exit flow, although it may not yield an adequate resolution at upstream infinity for certain applications such as cascades.

3.2 TRANSFORMATIONS BETWEEN PHYSICAL AND COMPUTATIONAL PLANE

The general transformation from the physical plane $[x, y]$ to the transformed plane has the form

$$\begin{bmatrix} \xi \\ \eta \end{bmatrix} = \begin{bmatrix} \xi(x,y) \\ \eta(x,y) \end{bmatrix} \tag{24}$$

The Jacobian matrix for this transformation is

$$J_1 = \frac{\partial(\xi,\eta)}{\partial(x,y)} = \begin{bmatrix} \frac{\partial\xi}{\partial x} & \frac{\partial\xi}{\partial y} \\ \frac{\partial\eta}{\partial x} & \frac{\partial\eta}{\partial y} \end{bmatrix} = \begin{bmatrix} \xi_x & \xi_y \\ \eta_x & \eta_y \end{bmatrix} \tag{25}$$

The inverse function or transformation of (24) is,

$$\begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} x(\xi,\eta) \\ y(\xi,\eta) \end{bmatrix} \tag{26}$$

with Jacobian matrix

$$J_2 = \frac{\partial(x,y)}{\partial(\xi,\eta)} = \begin{bmatrix} \frac{\partial x}{\partial\xi} & \frac{\partial x}{\partial\eta} \\ \frac{\partial y}{\partial\xi} & \frac{\partial y}{\partial\eta} \end{bmatrix} = \begin{bmatrix} x_\xi & x_\eta \\ y_\xi & y_\eta \end{bmatrix} \tag{27}$$

The Jacobian is then:

$$J = det[J] = x_\xi y_\eta - x_\eta y_\xi$$

Since $J_1 = J_2^{-1}$,

$$\begin{bmatrix} \xi_x & \xi_y \\ \eta_x & \eta_y \end{bmatrix} = J^{-1} \begin{bmatrix} y_\eta & -x_\eta \\ y_\xi & x_\xi \end{bmatrix} \tag{28}$$

we get the following relations:

$$\xi_x = \frac{y_\eta}{J} \tag{29a}$$

$$\eta_x = \frac{-y_\xi}{J} \tag{29b}$$

$$\xi_y = \frac{-x_\eta}{J} \tag{30a}$$

$$\eta_y = \frac{x_\xi}{J} \tag{30b}$$

Using the chain rule, partial derivatives of a given function f with respect to x and y are transformed as follows,

$$f_x = \frac{\partial f}{\partial x} = \frac{\frac{\partial(f,y)}{\partial(\xi,\eta)}}{\frac{\partial(x,y)}{\partial(\xi,\eta)}} = \frac{y_\eta f_\xi - y_\xi f_\eta}{J} \tag{31}$$

$$f_y = \frac{\partial f}{\partial y} = \frac{\frac{\partial(x,f)}{\partial(\xi,\eta)}}{\frac{\partial(x,y)}{\partial(\xi,\eta)}} = \frac{x_\xi f_\eta - x_\eta f_\xi}{J} \tag{32}$$

$$\begin{aligned} f_{xx} &= \frac{\partial}{\partial x}\left(\frac{y_\eta f_\xi - y_\xi f_\eta}{J}\right) \\ &= (y_\eta^2 f_{\xi\xi} - 2y_\xi y_\eta f_{\xi\eta} + y_\xi^2 f_{\eta\eta})/J^2 \\ &\quad + [(y_\eta^2 y_{\xi\xi} - 2y_\xi y_\eta y_{\xi\eta} + y_\xi^2 y_{\eta\eta})(x_\eta f_\xi - x_\xi f_\eta) \\ &\quad + [(y_\eta^2 x_{\xi\xi} - 2y_\xi y_\eta y_{\xi\eta} + y_\xi^2 x_{\eta\eta})(y_\xi f_\eta - y_\eta f_\xi)]/J^3 \end{aligned} \tag{33}$$

$$\begin{aligned} f_{yy} &= \frac{\partial}{\partial y}\left(\frac{-x_\eta f_\xi + x_\xi f_\eta}{J}\right) \\ &= (x_\eta^2 f_{\xi\xi} - 2x_\xi x_\eta f_{\xi\eta} + x_\xi^2 f_{\eta\eta})/J^2 \\ &\quad + [(x_\eta^2 y_{\xi\xi} - 2x_\xi x_\eta y_{\xi\eta} + x_\xi^2 y_{\eta\eta})(x_\eta f_\xi - x_\xi f_\eta) \\ &\quad + [(x_\eta^2 x_{\xi\xi} - 2x_\xi x_\eta x_{\xi\eta} + x_\xi^2 x_{\eta\eta})(y_\xi f_\eta - y_\eta f_\xi)]/J^3 \end{aligned} \tag{34}$$

$$\begin{aligned} f_{xy} &= [(x_\xi y_\eta + x_\eta y_\xi) f_{\xi\eta} - x_\xi y_\xi f_{\eta\eta}]/J^2 \\ &\quad + [x_\eta y_\eta x_{\xi\xi} - (x_\xi y_\eta + x_\eta y_\xi) x_{\xi\eta} + x_\xi y_\xi x_{\eta\eta}](y_\eta f_\xi - y_\xi f_\eta)/J_3 \\ &\quad + [x_\eta y_\eta y_{\xi\xi} - (x_\xi y_\eta + x_\eta y_\xi) y_{\xi\eta} + x_\xi y_\xi y_{\eta\eta}](x_\xi f_\eta - x_\eta f_\xi)/J^3 \end{aligned} \tag{35}$$

and so on.

Sufficient conditions for the transformations (26) to exist are given by the inverse function theorem that states: if the component functions of (24) are continuously differentiable at some point, say (x_1, y_1), and Jacobian matrix (25) is nonsingular at (x_1, y_1) then there exists a region M_1 about (x_1, y_1) such that the inverse function (26) exists and (28) holds for all $[(x, y)]$ in M_1. It is apparent that the theorem guarantees existence only in a local sense. For this reason component functions of (24) which posses even more desirable properties than those stated for the inverse function theorem are sought.

4 AN ELLIPTIC SCHEME AND APPLICATION

The basic idea of the transformations in section (3) is to let the component functions of (24) be solutions of an elliptic Dirichlet boundary value problem. An obvious choice is to require that $\xi(x, y)$ and $\eta(x, y)$ be either harmonic, subharmonic, or superharmonic. Harmonic functions obey a maximum principle, which states that the maximum and minimum values of the function must occur on the boundaries of the region D. Since no extremes occur within D, the first derivatives of the function will not simultaneously vanishes in D, and hence the Jacobian will not be zero due to the presence of an extremum. The maximum principle also guarantees uniqueness of the coordinate functions $\xi(x, y)$ and $\eta(x, y)$, and thus ensures that no overlapping of the boundaries will occur.

Now, let us take Laplace's equation as the generating elliptic system.

$$\xi_{xx} + \xi_{yy} = 0 \tag{36a}$$

$$\eta_{xx} + \eta_{yy} = 0 \tag{36b}$$

with the Dirichlet boundary conditions

$$\begin{bmatrix} \xi \\ \eta \end{bmatrix} = \begin{bmatrix} \xi_1(x,y) \\ \eta_1 \end{bmatrix}; [x,y] \in G_1 \tag{37a}$$

$$\begin{bmatrix} \xi \\ \eta \end{bmatrix} = \begin{bmatrix} \xi_2(x,y) \\ \eta_2 \end{bmatrix}; [x,y] \in G_2 \tag{37b}$$

where η_1 and η_2 are constants and $\xi_1(x,y)$ and $\xi_2(x,y)$ are specified monotonic functions on G_1 and G_2 respectively (fig. 6). That is, let $\xi(x,y)$ and $\eta(x,y)$ be harmonic in D. This generation system guarantees a one-to-one mapping for boundary-conforming curvilinear coordinate systems on general closed boundaries. Since all numerical computations are to be performed in a uniform rectangular transformed plane, the dependent and independent variables must be interchanged in (36). Using transformations (31) - (35), and knowing that coordinate lines in the transformed plane are constant, we get

$$\begin{aligned} \xi_{xx} &= [(y_\eta^2 y_{\xi\xi} - 2y_\xi y_\eta y_{\xi\eta} + y_\xi^2 y_{\eta\eta})x_\eta \\ &\quad + (y_\eta^2 x_{\xi\xi} - 2y_\xi y_\eta y_{\xi\eta} + y_\xi^2 x_{\eta\eta})(-y_\eta)]/J^3 \end{aligned} \tag{38}$$

$$\begin{aligned} \eta_{xx} &= [(y_\eta^2 y_{\xi\xi} - 2y_\xi y_\eta y_{\xi\eta} + y_\xi^2 y_{\eta\eta})(-x_\xi) \\ &\quad + (y_\eta^2 x_{\xi\xi} - 2y_\xi y_\eta y_{\xi\eta} + y_\xi^2 x_{\eta\eta})(y_\xi)]/J^3 \end{aligned} \tag{39}$$

$$\begin{aligned} \xi_{yy} &= [(x_\eta^2 y_{\xi\xi} - 2x_\xi x_\eta y_{\xi\eta} + x_\xi^2 y_{\eta\eta})x_\eta \\ &\quad + (x_\eta^2 x_{\xi\xi} - 2x_\xi x_\eta x_{\xi\eta} + x_\xi^2 x_{\eta\eta})(-y_\eta)]/J^3 \end{aligned} \tag{40}$$

$$\begin{aligned} \eta_{yy} &= [(x_\eta^2 y_{\xi\xi} - 2x_\xi x_\eta y_{\xi\eta} + x_\xi^2 y_{\eta\eta})(-x_\xi) \\ &\quad + (x_\eta^2 x_{\xi\xi} - 2x_\xi x_\eta x_{\xi\eta} + x_\xi^2 x_{\eta\eta})(y_\xi)]/J^3 \end{aligned} \tag{41}$$

and the transformed system has the following form

$$(Ax_{\xi\xi} + Bx_{\xi\eta} + Cx_{\eta\eta})(-y_\eta) + (Ay_{\xi\xi} + By_{\xi\eta} + Cy_{\eta\eta})(x_\eta) = 0 \qquad (42a)$$

$$(Ax_{\xi\xi} + Bx_{\xi\eta} + Cx_{\eta\eta})(y_\xi) + (Ay_{\xi\xi} + By_{\xi\eta} + Cy_{\eta\eta})(-x_\xi) = 0 \qquad (42b)$$

Where the coefficients are

$$A = x_\eta^2 + y_\eta^2 \qquad (43a)$$

$$B = -2(x_\xi x_\eta + y_\xi y_\eta) \qquad (43b)$$

$$C = x_\xi^2 + y_\xi^2 \qquad (43c)$$

or

$$-C_1 y_\eta + C_2 x_\eta = 0 \qquad (44a)$$

$$C_1 y_\xi - C_2 x_\xi = 0 \qquad (44b)$$

where

$$C_1 = Ax_{\xi\xi} + Bx_{\xi\eta} + Cx_{\eta\eta}$$

$$C_2 = Ay_{\xi\xi} + By_{\xi\eta} + Cy_{\eta\eta}$$

Since the Jacobian determinant is nonzero, a nontrivial solution of system (44) exists if $C_1 = C_2 = 0$, and we get the coupled system

$$Ax_{\xi\xi} + Bx_{\xi\eta} + Cx_{\eta\eta} = 0 \qquad (45a)$$

$$Ay_{\xi\xi} + By_{\xi\eta} + Cy_{\eta\eta} = 0 \qquad (45b)$$

with transformed boundary conditions

$$\begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} p_1(\xi, \eta_1) \\ p_2(\xi, \eta_1) \end{bmatrix}; [\xi, \eta_1] \in G_1^* \qquad (46a)$$

$$\begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} q_1(\xi,\eta_2) \\ q_2(\xi,\eta_2) \end{bmatrix}; [\xi,\eta_2] \in G_2^* \tag{46b}$$

The functions $p_1(\xi,\eta_1), p_2(\xi,\eta_1), q_1(\xi,\eta_2)$ and $q_2(\xi,\eta_2)$ are specified by the known shape of the inner and outer boundary. Assuming uniform spacing on the computational domain, the central difference equations approximating (45) at grid point (i,j) may be written,

$$\begin{aligned} &A(x_{i-1,j} - 2x_{i,j} + x_{i+1,j}) + \\ &B(x_{i-1,j-1} - x_{i-1,j+1} - x_{i+1,j-1} + x_{i+1,j+1})/4 + \\ &C(x_{i,j-1} - 2x_{i,j} + x_{i,j+1}) = 0 \end{aligned} \tag{47}$$

$$\begin{aligned} &A(y_{i-1,j} - 2y_{i,j} + y_{i+1,j}) + \\ &B(y_{i-1,j-1} - y_{i-1,j+1} - y_{i+1,j-1} + y_{i+1,j+1})/4 + \\ &C(y_{i,j-1} - 2y_{i,j} + y_{i,j+1}) = 0 \end{aligned} \tag{48}$$

that is, in general form

$$\begin{aligned} &A(s_{i-1,j} - 2s_{i,j} + s_{i+1,j}) + \\ &B(s_{i-1,j-1} - s_{i-1,j+1} - s_{i+1,j-1} + s_{i+1,j+1})/4 + \\ &C(s_{i,j-1} - 2s_{i,j} + s_{i,j+1}) = 0 \end{aligned} \tag{49}$$

where $s_{i,j}$ represents Cartesian coordinate x or y.

The discretized system (49) can be solved iteratively using a relaxation scheme of the form described previously, such as successive over-relaxation (SOR) or alternating direction implicit (ADI), until a specified convergence criterion is satisfied.

Adding forcing functions terms in eqs. (36) we get the Poisson system

$$\xi_{xx} + \xi_{yy} = P(\xi,\eta) \tag{50a}$$

$$\eta_{xx} + \eta_{yy} = Q(\xi, \eta) \tag{50b}$$

and applying the same transformation procedure this yields instead of (42)

$$Ax_{\xi\xi} + Bx_{\xi\eta} + Cx_{\eta\eta} = -J^2[P(\xi, \eta)x_\xi + Q(\xi, \eta)x_\eta] \tag{51a}$$

$$Ay_{\xi\xi} + By_{\xi\eta} + Cy_{\eta\eta} = -J^2[P(\xi, \eta)y_\xi + Q(\xi, \eta)y_\eta] \tag{51b}$$

As noted previously, the distribution of grid lines obtained by solution of the discretized form of equations (49) will tend to be equally spaced in the absence of boundary curvature because of the smoothing effect of the Laplacian, but will become more closely spaced over convex boundaries, and less so over concave boundaries, as illustrated in fig. 5a. For the first example in fig. 5a we have $\eta_{xx} > 0$ because of the convex curvature of the lines of constant η. Therefore it follows that $\eta_{yy} < 0$, and the spacing between the η - lines must increase with y. The η - lines thus will tend to be more closely spaced over a such convex boundary segment. For concave regions (the second example in fig. 5a) we have $\eta_{xx} < 0$, so that η_{yy} must be positive, and hence the spacing of the η - lines must decrease outward from this concave boundary. Considering eq. (50b), it follows that negative values of $Q(\xi, \eta)$ will tend to cause the coordinate line spacing to increase more rapidly outward from the boundary. In general, negative values of the control function $Q(\xi, \eta)$ will cause the η - lines to tend to move in the direction of decreasing η, while negative values of $P(\xi, \eta)$ in eq. (50a) will cause ξ - lines to tend to move in the direction of decreasing ξ. These effects are illustrated in fig. 5b for an η - line boundary. With the boundary values fixed, the ξ - lines here cannot change the intersection with the boundary. The effect of control function $P(\xi, \eta)$ in this case is to change the angle of intersection at the boundary, causing ξ - lines to "lean" in the direction of

decreasing ξ.

Generalizing, a negative value of the Laplacian of one of the curvilinear coordinates causes the lines on which that coordinate is constant to move in the direction in which that coordinate decreases. Positive values of the Laplacian naturally result in the opposite effect.

4.1 COORDINATE SYSTEM CONTROL

Control of the spacing of the coordinate lines on the body is accomplished through the input data, while the spacing of the coordinate lines in the field is controlled by varying the elliptic generating system for the coordinates. One procedure that has proven effective for some aerodynamic applications (Holst [17], Thompson, Thames and Mastin [16]), is to choose forcing functions $P(\xi,\eta)$ and $Q(\xi,\eta)$ as exponential terms

$$\begin{aligned} P(\xi,\eta) &= -\sum_{i=1}^{N} a_i sign(\xi-\xi_i) e^{(-c_i|\xi-\xi_i|)} \\ &- \sum_{j=1}^{M} b_j sign(\xi-\xi_i) e^{-d_j\sqrt{[(\xi-\xi_j)^2+(\eta-\eta_j)^2]}} \end{aligned} \tag{52}$$

$$\begin{aligned} Q(\xi,\eta) &= -\sum_{i=1}^{N} a_i sign(\eta-\eta_i) e^{(-c_i|\eta-\eta_i|)} \\ &- \sum_{j=1}^{M} b_j sign(\eta-\eta_i) e^{-d_j\sqrt{[(\xi-\xi_j)^2+(\eta-\eta_j)^2]}} \end{aligned} \tag{53}$$

The first terms in $P(\xi,\eta)$ and $Q(\xi,\eta)$ have the effect of attracting the $\xi =$ *constant* lines to the $\xi = \xi_i$ lines in eq. (52), and attracting $\eta =$ *constant* lines to $\eta = \eta_i$ lines in eq. (53). The second terms cause $\xi =$ *constant* lines to be attracted to the points (ξ_j, η_j) in eq. (52) with a similar effect on $\eta =$ *constant* lines in eq. (53). Introducing the *sign* function, the equations

(52) and (53) are no longer subharmonic or superharmonic, since the *sign* function causes a sign change on the right side when the attraction is to lines or points not on the boundaries. So, it is possible that too strong an attraction may cause the system to overlap and produce unusable grids. The use of the *sign* - changing function is used to cause attraction to both sides of a line or point in the field. Elimination of this function causes attraction on one side and repulsion on the other. If it is only desired that coordinate lines be concentrated near one boundary, (i.e., the body surface) then there is no need for the sign change, and the *sign* function can be eliminated. In this case the equations are subharmonic or superharmonic, and a maximum principle is in effect to prevent overlap. The effect of the amplitude factors in (52, 53) is shown in fig. 5c.

4.2 APPLICATION OF THE ELLIPTIC SCHEME

The elliptic grid generation scheme can be applied to both single and multiple bodies. Referring to fig. 6, assume that the body contour and the outer boundary transform, respectively, to the η - lines forming the lower and upper sides of the rectangular transformed plane; arbitrary cuts that join those boundaries map to the ξ - lines forming left and right sides of the transformed plane. Thus the left and right side vertical boundaries in the computational plane are coincident in the physical plane; the values of x and y coordinates are equal along these lines. The computational field size is $(JM-2)$ x $(IM-1)$. Boundary values are specified on $j=1$ and $j=JM$ for all $1<i<IM$. The line $j=1$ corresponds to the body surface in the physical plane while $j=JM$ corresponds to the outer boundary. The discretized form of the governing equation in the computational plane is given by (49). The re - entrant boundaries occur at $i=1$ and $i=IM$.

Since the values of x and y are equal along these lines, iteration is necessary along only one of them. For $i = 1$, the ξ - derivatives along this line can be approximated as

$$(x_\xi)_{i,j} = (x_{2,j} - x_{IM,j})/2 \tag{54a}$$

$$(x_{\xi\xi})_{i,j} = x_{2,j} - 2x_{1,j} + x_{IM-1,j} \tag{54b}$$

$$(x_{\xi\eta})_{i,j} = (x_{2,j+1} - x_{2,j-1} + x_{IM-1,j-1} - x_{IM-1,j+1})/4 \tag{54c}$$

for $2 < j < JM-1$. Similar expressions can be used for the derivatives of y. The set of simultaneous difference equations in $x_{i,j}$ and $y_{i,j}$ is solved by point SOR iteration in the present numerical experiments. The same idea may be extended to multiple bodies. In this case we introduce additional cuts that connect bodies and apply a computational procedure similar to that for the single body. For two-component bodies the transformed boundary conditions are given as follows (fig. 7)

$$\begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} p_1(\xi,\eta_1) \\ p_2(\xi,\eta_1) \end{bmatrix}; [\xi,\eta_1] \in G_1^* \tag{55a}$$

$$\begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} q_1(\xi,\eta_2) \\ q_2(\xi,\eta_2) \end{bmatrix}; [\xi,\eta_2] \in G_2^* \tag{55b}$$

$$\begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} p_3(\xi,\eta_1) \\ p_4(\xi,\eta_1) \end{bmatrix}; [\xi,\eta_1] \in G_7^* \tag{56a}$$

$$\begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} q_3(\xi,\eta_2) \\ q_4(\xi,\eta_2) \end{bmatrix}; [\xi,\eta_2] \in G_8^* \tag{56b}$$

The functions $p_1, p_2, q_1, q_2, p_3, p_4, q_3$ and q_4 are specified by the known shape of the contours G_1, G_2, G_7 and G_8 and the specified distribution of ξ thereon. Re-entrant boundaries occur on G_3^*, G_4^*, G_5^* and G_6^*. Derivative approximations on these boundaries are determined using a procedure similar to that given by (54). In the present study we generate grids about single and

multiple airfoils by use of equations (45) and (51) on uniform grids in the computational plane following the procedure of Thompson, Thames and Mastin [16].

Fig. 8 shows an O-grid about NACA 0012 airfoil generated by the elliptic scheme with clustering functions $P(\xi, \eta)$ and $Q(\xi, \eta)$ given by eqs.(52, 53). The grid size is 61 x 28 and radius of the outer boundary 3.0 x chord. The attraction is applied for five coordinate lines near the airfoil surface and leading and trailing edges. The initial guess is determined using the average values of four boundary points. For a fixed acceleration parameter, a convergent solution was obtained after 99 iterations and 48.561 seconds of CPU time on the *CDC 750/175*. The grid in fig. 9 is generated using the elliptic scheme (45). The grid size is 49 x 16 and radius of the outer boundary 1.2 x chord. The initial guess is determined as the solution of the parabolic scheme (68, 69). In this case a convergent solution was obtained in 1.971 seconds of CPU time. Good clustering of the coordinate lines at leading and trailing edges of the airfoil is evident.

In fig. 10 and fig. 11 C-type and H-type grids, respectively, are shown. Both grids are generated using the elliptic system (45). The initial guess is determined in the same way as for the O-type grid. It can be seen in all three cases that the system (45) provides good smoothness of the grid lines, but does not provide clustering of the grid lines at the body surface.

Fig. 12 is an O-grid about two NACA 0012 airfoils with flap angle of 25 degrees. Grid size is 69 x 20 and radius of the outer boundary 4.0 x chord. Attraction is applied for six coordinate lines near the airfoils and at the leading and trailing edges of both airfoils. The initial guess is specified using the average values of four boundary points. Convergence of the solution, with constant acceleration factor, was obtained after 275

iterations and 105.702 seconds of CPU time. A detailed grid distribution near the airfoils is given in fig. 13.

In all numerical experiments it was observed that the efficiency of the elliptic scheme depends on the accuracy of the initial guess. Using the average values of four boundary points or linear/exponential projection of boundary points offers the possibility of generating good starting iterates. Experience indicates that solution for both single and multiple airfoils, is sensitive to the initial guess and distance to the outer boundary and in some cases a solution may not be obtained. The solution often converges slowly.

5 PARABOLIC SCHEME AND APPLICATION

5.1 GENERAL OBSERVATIONS

In the present study we examine the feasibility of using parabolic partial differential equations for grid generation by considering the pair of model equations

$$a(\xi,\eta)x_\eta = b(\xi,\eta)x_{\xi\xi} + c(\xi,\eta)V_x(\xi,\eta) + d(\xi,\eta) \tag{57}$$

$$a(\xi,\eta)y_\eta = b(\xi,\eta)y_{\xi\xi} + c(\xi,\eta)V_y(\xi,\eta) + d(\xi,\eta) \tag{58}$$

where a, b and c can be constants or some functions of (ξ,η); (x,y) denote the coordinates of the physical domain, (ξ,η) the computational domain, and (V_x, V_y) source terms. The physical and computational domains are shown in fig. 14. Unlike the elliptic scheme, in the present approach the inner boundary represents an initial condition, and the outer boundary represents a constraint that effects the j-coordinate line distribution. The

point distribution on remaining segments is given and represents boundary conditions on left and right sides of the computational plane. Equations (57,58) can be discretized using backward differencing in the (timelike) η-coordinate. Given the initial values of x and y at $\eta = 0$ this leads to a tridiagonal system to be solved for each increment of η.

The initial values are specified as

$$x(\xi, 0) = x_0(\xi)$$

$$y(\xi, 0) = y_0(\xi) \tag{59}$$

where $x_0(\xi)$ and $y_0(\xi)$ are the coordinates of the body surface. In order to examine the effect of the source terms , let us set $b = d = 0$ so that

$$\frac{\partial x(\xi, \eta)}{\partial \eta} = \frac{c}{a} V_x(\xi, \eta) \tag{60}$$

$$\frac{\partial y(\xi, \eta)}{\partial \eta} = \frac{c}{a} V_y(\xi, \eta) \tag{61}$$

From (60,61) we see that as η increases, the change of x and y is determined by V_x and V_y. This implies that V_x and V_y should be specified in such a way that x and y change in a desired direction and amount. The role of $x_{\xi\xi}$ and $y_{\xi\xi}$ in eqs. (57,58) may be considered as smoothing the grid intervals in the ξ direction. V_x and V_y can be approximated in (60,61) by using linear or polynomial interpolation between the inner and outer boundaries. The orthogonality of the grid lines may be controlled by introducing a "fictitious" outer boundary, and grid spacing in both the ξ and η directions can be controlled by discretization of the governing equations (57,58) on a nonuniform mesh in the computational plane.

5.2 NUMERICAL DEVELOPMENT OF PARABOLIC SCHEME

In most coordinate transformations used for computational analyses the grid spacing in both the ξ and η directions on the computational domain is set to unity. However, this restriction is not necessary. When deriving generation equations, we can assume that grid spacings are quasi- uniform on the computational domain. Once a grid is generated, it may be used for flow calculations as if generated on a uniformly spaced grid on the computational domain. Using Taylor series expansion on a nonuniform mesh and referring to fig. 15a, derivatives can be approximated as follows

$$f_x = \frac{f_{i+1,j} - f_{i,j}}{\Delta R} \quad or \quad f_x = \frac{f_{i,j} - f_{i-1,j}}{\Delta L}$$

$$f_y = \frac{f_{i,j+1} - f_{i,j}}{\Delta U} \quad or \quad f_y = \frac{f_{i,j} - f_{i,j-1}}{\Delta D}$$

$$f_{xx} = \frac{2}{\Delta R + \Delta L}\left(\frac{f_{i+1,j} - f_{i,j}}{\Delta R} + \frac{f_{i-1,j} - f_{i,j}}{\Delta L}\right)$$

$$f_{yy} = \frac{2}{\Delta D + \Delta U}\left(\frac{f_{i,j+1} - f_{i,j}}{\Delta U} + \frac{f_{i,j-1} - f_{i,j}}{\Delta D}\right)$$

$$f_{xy} = \frac{f_{i+1,j+1} - f_{i+1,j-1} - f_{i-1,j+1} + f_{i-1,j-1}}{(\Delta L + \Delta R)(\Delta D + \Delta U)}$$

and the governing equations (57,58) can be discretized as

$$\frac{a(x_{i,j} - x_{i,j-1})}{\Delta D} = \frac{2b}{(\Delta L + \Delta R)}\left[\frac{x_{i-1,j} - x_{i,j}}{\Delta L} + \frac{x_{i+1,j} - x_{i,j}}{\Delta R}\right] + c\frac{XB_{i,JMAX} - x_{i,j}}{\Delta O} + d \qquad (62)$$

$$\frac{a(y_{i,j} - y_{i,j-1})}{\Delta D} = \frac{2b}{(\Delta L + \Delta R)}\left[\frac{y_{i-1,j} - y_{i,j}}{\Delta L} + \frac{y_{i+1,j} - y_{i,j}}{\Delta R}\right] + c\frac{YB_{i,JMAX} - y_{i,j}}{\Delta O} + d \qquad (63)$$

In order to determine coefficients a, b and c and to explain the proposed method of controlling grid spacing the idea is to relate the model equations to some known set of equations in the computational plane. For this purpose, the grid generation of elliptic type (45), is selected. In fact, the procedure that we want to apply is essentially equivalent to deriving the elliptic difference equations (45) on the nonuniform grids of the computational domain in which the m-th grid line in the η-direction on a uniform grid is moved toward the $(m-1)$-th grid line. Using the above procedure, an arbitrary number of grid lines may be moved in both the ξ and η directions. By denoting the distance ratios by F_i and g_j in the ξ and η directions, respectively, (fig. 15b), the difference equations that yield the grid spacing can be written as

$$\begin{aligned}
&\frac{2A}{F_i + F_{i-1}}\left[\frac{s_{i-1,j} - s_{i,j}}{F_{i-1}} + \frac{s_{i+1,j} - s_{i,j}}{F_i}\right] + \\
&B\left[\frac{s_{i+1,j+1} - s_{i+1,j-1} - s_{i-1,j+1} + s_{i-1,j-1}}{(F_{i-1} + F_i)(g_{j-1} + g_j)}\right] + \\
&\frac{2C}{g_j + g_{j-1}}\left[\frac{s_{i,j-1} - s_{i,j}}{g_{j-1}} + \frac{s_{i,j+1} - s_{i,j}}{g_j}\right] = 0
\end{aligned} \tag{64}$$

We then have elliptic PDE's (45) discretized on a quasi-uniform mesh in (ξ, η).

Now, a marching grid generation equation scheme may be obtained from (64) by replacing the coordinates $x_{i,j+1}$ and $y_{i,j+1}$ by known values $XO_{i,j+1}$ and $YO_{i,j+1}$ obtained from linear interpolation between coordinates of the outer and inner boundaries. The ratio g_j can be relaxed in practice i.e., it may be gradually changed along the grid line in some functional dependence.

In the simplest case, when we do not force orthogonality on the boundaries, coordinates XO and YO are set to the known values ($XB_{i,JMAX}$,

$YB_{i,JMAX}$) at the outer boundary. In general, however, the marching grid generation equations for both the x and y coordinate may be written in the form:

$$\frac{2A}{F_i+F_{i-1}}[\frac{s_{i-1,j}-s_{i,j}}{F_{i-1}}+\frac{s_{i+1,j}-s_{i,j}}{F_i}]+$$
$$\frac{2C}{g_{j-1}+G_j}[\frac{-s_{i,j}}{g_{j-1}}-\frac{s_{i,j}}{G_j}]$$
$$=-B[\frac{SO_{i+1,j+1}-SO_{i-1,j-1}-s_{i+1,j-1}+s_{i-1,j-1}}{(F_{i-1}+F_i)(g_{j-1}+G_j}]$$
$$-\frac{2C}{g_{j-1}+G_j}[\frac{s_{i,j-1}}{g_{j-1}}+\frac{SO_{i,j+1}}{G_j}] \qquad (65)$$

where:

$$G_j=g_j+g_{j+1}+....g_{JMAX-1}=\eta_{MAX}-\eta_j$$

$$g_{j-1}=\eta_j-\eta_{j-1}\ ;\ F_i=\xi_{i+1}-\xi i\ ;\ F_{i-1}=\xi_i-\xi_{i-1}$$

G_j represents the distance between grid lines (i,j) and (i,JO) on the computational plane and $SO_{i,j}=XO_{i,j}$ or $YO_{i,j}$. In order to establish a relation between discretized model parabolic system (62,63) and system (65), let us compare those two systems. Separating known and unknown values we get

$$-(\frac{b}{\Delta L}+\frac{b}{\Delta R})\frac{2s_{i,j}}{\Delta L+\Delta R}-(\frac{a}{\Delta D}+\frac{c}{\Delta O})s_{i,j}+\frac{b}{\Delta R}\frac{2s_{i-1,j}}{(\Delta R+\Delta L)}+$$
$$\frac{b}{\Delta R}\frac{2s_{i+1,j}}{(\Delta R+\Delta L)}=-c\frac{SB_{i,JMAX}}{\Delta O}-a\frac{s_{i,j-1}}{\Delta D}-d \qquad (66)$$

$$-(\frac{A}{F_{i-1}}+\frac{A}{F_i})\frac{2s_{i,j}}{(F_i+F_{i-1})}-(\frac{C}{g_{j-1}}+\frac{C}{G_j})\frac{2s_{i,j}}{G_j+g_{j-1}}+\frac{A}{F_{i-1}}\frac{2s_{i-1,j}}{(F_i+F_{i-1})}+$$
$$\frac{A}{F_i}\frac{2s_{i+1,j}}{(F_i+F_{i-1})}=-\frac{2C}{g_{j-1}+G_j}[\frac{SO_{i,j+1}}{G_j}+\frac{s_{i,j-1}}{g_{j-1}}]-$$
$$B[\frac{SO_{i+1,j+1}-SO_{i-1,j+1}-s_{i+1,j-1}+s_{i-1,j-1}}{(F_i+F_{i-1})(g_{j-1}+G_j)}] \qquad (67)$$

In both equations the right sides are known values. It can be seen that they have the same form. So, we can establish a relation between coefficients of eqs. (62,63) and (65) as follows

$$b = A$$

$$a = c = \frac{2C}{(G_j + g_{j-1})}$$

$$d = B[\frac{SO_{i+1,j+1} - SO_{i-1,j+1} - s_{i+1,j-1} + s_{i-1,j-1}}{(F_i + F_{i-1})(G_j + g_{j-1})}]$$

Where the coefficients A, B and C are given by (43). In fact, system (65) represents a 2 x 2 block tridiagonal system that can be evaluated as follows: defining

$$\alpha = \frac{2A}{F_{i-1}(F_i + F_{i-1})} \; ; \; \gamma = \frac{2A}{F_i(F_i + F_{i-1})}$$

$$\beta = \frac{-2A}{F_i + F_{i-1}}(\frac{1}{F_i} + \frac{1}{F_{i-1}}) - \frac{2C}{G_j + g_{j-1}}(\frac{1}{G_j} + \frac{1}{g_{j-1}})$$

$$D_x = -B[XO_{i+1,j+1} - XO_{i-1,j+1} - x_{i+1,j-1} + x_{i-1,j-1}]/[(F_i + F_{i-1})(G_j + g_{j-1})] - \frac{2C}{G_j + g_{j-1}}[x_{i,j-1}/g_{j-1} + XO_{i,j+1}/G_j]$$

$$D_y = -B[YO_{i+1,j+1} - YO_{i-1,j+1} - y_{i+1,j-1} + y_{i-1,j-1}]/[(F_i + F_{i-1})(G_j + g_{j-1})] - \frac{2C}{G_j + g_{j-1}}[y_{i,j-1}/g_{j-1} + YO_{i,j+1}/G_j]$$

the governing equation in the x-direction becomes

$$\alpha x_{i-1,j} + \beta x_{i,j} + \gamma x_{i+1,j} = D_x \tag{68}$$

and similarly in the y-direction:

$$\alpha y_{i-1,j} + \beta y_{i,j} + \gamma y_{i+1,j} = D_y \tag{69}$$

Equations (68) and (69) are solved simultaneously for all grid points on the j-th grid line by using a tridiagonal solver for each $x_{i,j}$ and $y_{i,j}$. The solution starts with $j = 2$, (next to the inner boundary) and marches until $j = JMAX - 1$, (next to the prescribed outer boundary grid line). The coefficients of A, B and C are calculated using the coordinates of adjacent grid lines that are already generated, and the first derivatives that appear in them can be approximated by differencing as,

$$x_\xi = \frac{x_{i+1,j-1} - x_{i-1,j-1}}{F_i + F_{i-1}} \tag{70a}$$

$$y_\xi = \frac{y_{i+1,j-1} - y_{i-1,j-1}}{F_i + F_{i-1}} \tag{70b}$$

$$x_\eta = \frac{XO_{i,j+1} - x_{i,j-1}}{g_{j-1} + G_j} \tag{71a}$$

$$y_\eta = \frac{YO_{i,j+1} - y_{i,j-1}}{g_{j-1} + G_j} \tag{71b}$$

The non-uniform grid spacing terms in the above scheme have effects similar to the spacing control terms in the exponential functions introduced by Thompson, Thames and Mastin [3] for the elliptic scheme. The value of F or g and the distance between the adjacent grids satisfy approximately a linear relationship. Therefore, if a grid is generated by using a known set of F and g for all the grid intervals, and if a different spacing distribution is desired in the next grid generation the values of F and g that satisfy the desired grid spacing can be found by the linear relationship between F and g and the grid spacing in the previous calculation.

5.3 GRID SPACING CONTROL

In computation of viscous flows or boundary layers, orthogonality of grid lines near the boundary surface is desirable to represent all normal deriva-

tives simply and accurately. To obtain this, the angular orientation of grid lines may be controlled by using a modified outer boundary grid to determine the right hand side source terms. For example, if the grid is shifted in such a way that the new outer boundary points lie on lines which are perpendicular to the body surfaces, the grid lines will be away from the inner surface approximately orthogonally.

Let us first consider the coordinates $(XO_{i,3}, YO_{i,3})$, corresponding to the grid line $j = 2$. Referring to fig. 16 a straight line AA is extended outward normal to the airfoil surface from the grid $(i, 1)$. A circular arc CC that passes through the point $(XB_{i,JMAX}, YB_{i,JMAX})$ on the outer boundary and has its center at $(i, 1)$ is drawn. The coordinates ($XO_{i,3}$ and $YO_{i,3}$) are set to those of the intersection S of the two lines AA and CC.

In the present work, the modified outer boundary is computed by first calculating the slope of the body surface and the distance from the body surface points to corresponding outer boundary points. The modified boundary is scaled so that it lies at the same distance from the body surface as the actual outer boundary. This is needed to obtain the desired clustering or stretching of grid lines. Since the parabolic algorithm generates the grid lines progressively toward the outer boundary the modified grid is gradually shifted back to the desired outer boundary point distribution. In this manner, angle control is maintained at the body surface, and a smooth transition can be made to the outer boundary.

5.4 EFFECT OF THE COEFFICIENTS ON GRID GENERATION

Here we explore the possibility of generating grids with different values of coefficients in (68,69). We consider the cases:

- CASE 1:

$$A = A(\xi, \eta), B = B(\xi, \eta), C = C(\xi, \eta)$$

- CASE 2:

$$A = A(\xi, \eta), B = 0, C = C(\xi, \eta)$$

- CASE 3:

$$A = A(\xi, \eta), B = 0, C = 1$$

- CASE 4:

$$A = 0, B = B(\xi, \eta), C = C(\xi, \eta)$$

Case 1 represents the method applied in the present study. *Cases 2* and *3* produce reasonable grids (see fig. 17a and fig. 17b, respectively), but in these cases smoothness of the grid lines is poor. *Case 4* is interesting because it provides a completely explicit grid generation method in the η direction. Grids from case 4 are given in fig. 17c and we can see that they are very similar to that obtained for case 1. Other combinations of coefficients may not provide grids because boundary information is insuff ient or due to the changing type of the governing equations.

5.5 PRACTICAL APPLICATION AND RESULTS

- SINGLE AIRFOILS

In the present work, all three commonly used types of grids - H-type, O-type and C-type - are generated. The complete procedure is given for O-type of grids, and the other two briefly explained.

First of all, let us consider the transformation of a two-dimensional doubly connected region D bounded by two closed contours onto a rectangular

region D, as shown in fig. 18 (the body contour and outer boundary are transformed, respectively, to the constant η - lines forming the bottom and top sides of the transformed region). Let G_1 represent the inner boundary, G_2 outer boundary, D physical plane, and D_1 transformed plane. In order to connect G_1 and G_2 it is necessary to make an arbitrary cut between G_3 and G_4.

Conceptually this can be viewed as an opening of the field at the cut and then a deformation into a rectangle. Now, let G_1 map onto G_1^*, G_2 onto G_2^*, G_3 onto G_3^* and G_4 onto G_4^*. G_1^* and G_2^* are constant η-lines in the transformed plane. Contours G_3 and G_4 which connect the contours G_1 and G_2 are coincident in the physical plane and form left and right sides of the computational plane with x and y values given as pre-assigned data.

To summarize, at this point we are given "initial" conditions on the inner boundary, a constraint condition for the outer boundary and boundary condition on cut as follows

$$\begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} f_1(\xi,\eta) \\ f_2(\xi,\eta) \end{bmatrix}; [\xi,\eta] \in G_1^* \tag{72}$$

$$\begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} f_3(\xi,\eta) \\ f_4(\xi,\eta) \end{bmatrix}; [\xi,\eta] \in G_2^* \tag{73}$$

$$\begin{bmatrix} x \\ y \end{bmatrix} = [f_5[(\xi_1,\eta),(\xi_2,\eta)]]; [\xi,\eta] \in G_3^*, G_4^* \tag{74}$$

The functions $f_1(\xi,\eta), f_2(\xi,\eta), f_3(\xi,\eta), f_4(\xi,\eta)$ are determined using the known shape of the inner and outer boundaries G_1 and G_2, and the specified distribution of ξ thereon, while the function f_5 is determined by the shape of the cut, and distribution of ξ lines thereon.

The boundary fitted coordinate system generated by solving (68, 69) has a constant η line coincident with each boundary in the physical plane.

The ξ = constant lines may be specified as desired around boundaries, since the assignment of the ξ - values to the (x, y) boundary points via the functions f_1, f_2, f_3 and f_4 is arbitrary. Control of the radial spacing of the ξ = constant lines is very important because of its influence on non-uniform grid spacing terms in the finite difference equations (65). Knowing boundary conditions (73, 74) and initial conditions (72) we can calculate the grid by marching from inner to outer boundary.

Figures 19 and 20 show O-type grids about *NACA 0012* and *Karman-Trefftz* airfoils generated in this way. The grid points on the outer boundary are equally spaced along a circle with radius of 1.5 x chord length. Orthogonality control is applied in the vicinity of the airfoil surface. Grid size for NACA 0012 is 61 x 28 (CPU time 0.258 sec.) and for Karman - Trefftz airfoil 41 x 16 (CPU time 0.189 sec.). Fig. 21 shows an O-grid generated about a NACA 0012 airfoil with strong clustering in vicinity of the airfoil surface. This type of coordinate line clustering can be considered for viscous and layer problems.

Figure 22 shows H-type grids without orthogonality at the airfoil surface and generated in the same way. The horizontal line of symmetry is used as initial data. Distribution of points on the outer boundary is the same as on the line of symmetry. Essentially, the grid is equivalent to that obtained by an algebraic grid generation scheme applied by Johnson [34] for transonic flow calculation about a single airfoil in channel conditions. The grid size is 41 x 20 and solution is obtained after 0.126 sec of CPU time. The grids above the airfoil were first generated starting from the airfoil surface to the top boundary, and then grids below the airfoil were generated by the same procedure. The best grid resolution was obtained by specifying a nonuniform spacing function F in the computational plane corresponding

to Δx in the physical plane. Fig. 23 shows H-grids about a NACA 0012 airfoil with strong clustering and orthogonality of grid lines in the vicinity of the airfoil surface. Obviously, both grids can be used for flow analysis about an airfoil in wind-tunnel conditions.

Fig. 24 indicates C-type grids similarly generated. In this case the wake position was taken to define the initial condition for resolution of grid lines from the trailing edge of the airfoil in the downstream direction. The outer boundary is uniformly discretized and the point distribution downstream from the trailing edge is identical to that on a wake. The grid size is 73 x 16 and CPU time 0.231 sec.

Opening the field at the cut (fig. 25) the two members of the pair of segments forming the branch cut are similarly directed in the transformed region, and consequently points located at a vertical distance below the segment 1-2, at a horizontal distance to the left of point 2, coincide with points at the same vertical distance above the segment 4-3, at the same horizontal distance to the left of point 3. In this case, ξ varies to the right on the upper side of the cut, but to the left on the lower side. The direction of variation of η also reverses at the cut, so that although the type and shape of both lines are continuous across the cut, the direction of variation reverses there.

In all cases shown, for O-type and C-type grids the nonuniform spacing functions F were set to $ds = \sqrt{\Delta x_i^2 + \Delta y_i^2}$ on the $j-1$ grid line, while nonuniform spacing of the η coordinate was enforced by use of trigonometric sine functions to produce clustering in desired regions.

- MULTIPLE AIRFOILS

The basic ideas and procedure introduced for single airfoils can be extended to regions containing more than one body, i.e., multiconnected or multi - body configurations. An example of the transformation for two airfoils is given in figure 26. The bodies are connected with one arbitrary cut and an additional arbitrary cut joining one of the body contours to the outer boundary. The physical plane contours $G_1 - G_8$ map respectively onto the contours $G_1^* - G_8^*$ in the transformed plane. The conceptual opening here is as follows: the pairs of segments (1-2,7-8) and (3-4,5-6) are the branch cuts, which form re-entrant boundaries in the transformed plane. In this case, points outside the right side of the transformed region coincide with points inside the left side, and vice versa. The coordinate type and direction are continuous across the cut. Points below the bottom segment 3-4, to the left of point 4, coincide with points above the segment 5-6 to the right of point 5 in the transformed plane. As those arbitrary cuts represent pre-assigned data for grid generation in the physical plane, there are a number of other possibilities for placement of the two cuts on the boundary of the transformed region.

In the present work, the procedure was applied to two NACA 0012 airfoils with flap at different angles. Examples with flap angle at 0 degrees and 25 degrees are shown in figures 27 and 28, respectively. Figure 29 gives more detailed grid line distribution for two NACA 0012 airfoils with flap angle of 25 degrees. The grid size is 87 x 30 and CPU time 0.623 sec. The results show very good resolution of grid lines and clustering effects of the nonuniform spacing terms.

• APPLICATION TO ARBITRARY CONFIGURATIONS

The method is, with the same basic ideas, applied further to cascade grids and configurations with two circles and a circle with an airfoil.

The first case demonstrates the ability of the parabolic method to satisfy outer boundary conditions a very small distance from the inner boundary. In this case the outer boundary was formed midway between neighboring airfoils. In fact, this cascade grid represents a composite system that can be extended upstream and downstream to infinity by two independent Cartesian systems similar to that given by Eiseman [7]. The main advantages of the composite system are that it can be used for highly - staggered, closely - spaced airfoils and avoids severe mesh distortion or growth that would occur at higher upstream or downstream extensions of the basic O-grids. The mesh in fig. 30a covers everything except triangular regions at the corner points that result from the demand for nonsingularity. Obviously, this system is to be applied only when these regions are sufficiently small and/or are located in places where the solution is slowly varying. An alternative approach is to relax the demand for nonsingularity and fill in the uncovered regions as in fig. 30b.

The configuration with two circles (fig. 31) shows the clustering ability of the method. Essentially the grids of this problem have very similar behavior to the example using an elliptic generator given by Thompson, Thames and Mastin [16].

The configuration containing circle and airfoil shapes (fig. 32) indicate the ability of the parabolic scheme to handle grid generation in a field with multiple bodies having very different geometry. The example can be applied for grid generation about an arrow wing - body configuration.

5.6 COMPUTATIONAL EFFICIENCY

The parabolic scheme produces viable grids in one outward "iteration" sweep. It uses a tridiagonal solver that is very time efficient. In order to investigate the computational efficiency of the parabolic scheme as a function of grid size we compare results with those obtained by the elliptic scheme with clustering functions. Results are given in Table 1 and show that the parabolic scheme is highly efficient. Morever it exhibits very little dependence on the distance of the outer boundary. In all cases for a single NACA 0012 airfoil CPU time includes forming the inner boundary, outer boundary and grid generation. In all other cases CPU time is for forming of the outer boundary and grid generation. All presented examples were run on the *CDC 170/750 Dual Cyber Computer* of the University Texas at Austin.

5.7 LIMITATIONS OF PARABOLIC SCHEME

We find that the most important and the most sensitive factors influencing the method are the distribution of points on cuts and specification of the nonuniform spacing terms F_i and g_j. In order to obtain good resolution of grid lines in the cut region for varying nonuniform spacing terms F_i and g_j in (68, 69) with j, the F_i and g_j should be given as functions of the point distribution. In this case clustering of the grid lines in the entire physical field is controlled by grid point distribution on the cuts. Otherwise we can have excessive skewness of the grids near the cuts. If we use constant values of F_i and g_j, to obtain different clustering of the grid lines we need several sweeps through the computational plane changing F_i and g_j in each sweep.

In application to multiple bodies we have additional cuts that connect

the bodies. In the case of asymmetric bodies we found that uniform spacing of points on cuts is most appropriate, since otherwise overlapping may occur. For bodies symmetrically situated in the physical plane, these cuts are not critical and the point distribution on them may be arbitrary.

5.8 CLUSTERING OF POINTS IN GIVEN REGIONS

The distribution and clustering of points along the airfoil contour, cuts,wake and "inlet" and "outlet" of H-type grids are controlled using a trigonometric sine function. We have two known parameters; the coordinate location of a point and the number of grid points that will be located up to the known point. The selected function is

$$y = L[x + FAC(N\pi x)] \tag{75}$$

where x linearly varies from zero to the length of the interval, FAC is a factor that determines the amount of clustering, N is 1 or 2 (for $N = 1$ we have a single and for $N = 2$ a double sine wave), and L is the physical length of the interval.

6 CONCLUSIONS

The present work shows that high quality grids can be generated by a marching solution of parabolic partial differential equations. The parabolic differencing scheme has been implemented for the generation of two-dimensional grids about single and multiple bodies. The method has demonstrated good spacing and angle control, and ability to generate smooth grids for finite difference computations in CFD.

The approximate orthogonality of the grids is enforced by introducing a fictitious outer grid contour. This shifted outer grid contour is computed by first computing the the slope of the airfoil surface and the distance from inner boundary points to corresponding outer boundary points.

On the basis of comparison of results with those obtained by an elliptic scheme, the method is demonstrated to have high computational efficiency and gives good resolution of grid lines. The present method appears stable.

Part II
POTENTIAL AIRFOIL ANALYSIS AND DESIGN

7 INTRODUCTION

The most popular and practical applicable methods for potential airfoil analysis and design are so called surface singularity or panel methods. The basic ideas of the panel methods introduced in the solution of arbitrary potential flow problems involve combining classical potential theory with contemporary numerical techniques. The classical theory provides a means to reduce flow problems to a surface integral equation relating boundary conditions to an unknown singularity distribution. Numerical techniques are then used to calculate an approximate solution of the integral equations. The procedure involves representing flow boundaries by surface elements (panels) on which potential flow singularities are distributed. The physical boundary condition is that the normal velocity is specified on all surfaces so that the mathematical formulation becomes a Neuman problem for Laplace's equation.

For a given geometry of an airfoil section the potential flow analysis methods provide surface velocity and pressure distribution and thus the lift, the pitching moment and other aerodynamic characteristics. The methods should be reliable and accurate because they are used as the first step in the

design of new wing sections. Generally, exact (i.e., conformal mapping) and surface singularity methods provide a good means for single airfoil potential flow analysis and design. However, for multicomponent airfoils exact methods produce mainly test cases, and the only practical design methods are based on surface singularity techniques. In a design mode we usually specify velocity or pressure distribution. Therefore, the performance of the system specified by design or inverse solution gives the geometry that will produce that performance. The inverse design is basically an iterative process that provides adjustments to both designed geometry and the required surface velocities at each iteration. So, we can include different constraints that frequently occur in the design of airfoil sections.

7.1 REVIEW OF POTENTIAL METHODS

The most widely used panel method is that of Hess and Smith [35] based on distribution of sources and sinks on the airfoil surface combined with a vorticity distribution to generate circulation. Improved solution based on a higher-order panel formulation was developed by Hess [36, 37, 38]. The main idea in development of higher order panel methods was to reduce panel density for a given solution accuracy and to reduce computing cost. However these advantages have not been achieved. For example, Maskew [39] demonstrates that a low-order panel method based on piecewise constant doublet and source singularity applied to general configurations for comparable density of control points give comparable accuracy to higher-order solutions. Bristow and Grose[40] introduce panel methods based on Green's third identity with constant or linear source distribution and linear vorticity distribution on flat panels and demonstrate very efficient inverse and mixed analysis/design for single and multicomponent airfoils. For two-

dimensional configurations a very popular method is the stream function approach based on constant or linear distribution of vorticity. For example, Ormsbee and Chen [41] use this method for multi-element airfoil design and use third-order Langrangian interpolation to determine surface points. Further, Kennedy and Marsden [42] introduce an additional control point a finite distance from the trailing edge to satisfy the Kutta condition and determine surface points using cubic spline or linear projection. This type of Kutta condition provides reduction in panel density and gives very good results. The method is also used as a starting solution in design of single airfoils in viscous incompressible flow by Dutt and Sreekanth [43] and for analysis and design of single airfoils in transonic flow by Greff and Mantel[44]

Instead of superposition of potentials due to surface singularities, a different type of boundary integral equation can be obtained by applying an appropriate Green's formula and fundamental solution to relate integrals over the interior of the domain to integrals on the boundary. Then, introducing a finite element expansion on a discretization of the boundary domain, an approximate solution of the boundary integral equations can be obtained. This procedure has been termed the ***boundary element method*** (Brebbia [45]). The method has been, with theoretical development, successfully applied to lifting airfoil calculations by Carey and Kim [46]. The major conceptual difference between panel and boundary element methods is; *the panel methods are based on superposition of surface singularities with discrete satisfaction of boundary flux conditions while in the boundary element methods one uses a finite element expansion and a discrete approximation of the boundary integral equations.*

Conformal mapping techniques have been successfully applied by Halsey

[9] for analysis of multi-element airfoils. Any number of airfoils are transformed to the same number of circles by successive application of a method for mapping a single body to a unit circle. Then, the flow about multiple circles is analysed using multiply-reflected doublets and vortices. Saddhoo and Hall [47] use the method of images and calculate the analytical solution for an inviscid incompressible flow past four arbitrary circles which are then conformally mapped onto airfoil sections.

In the present work we use the stream function approach of Kennedy and Marsden [42]. However, in the design mode we calculate the actual position of the trailing edge and then use linear interpolation to determine surface points. In this way it is possible to avoid saw-shaped airfoils that can result from propagation of small errors due to selection of additional trailing point as the actual trailing edge of the airfoil. The method is compared with several panel methods in analysis mode and demonstrated approximately to be of second order accuracy.

7.2 GENERAL CONSIDERATIONS

The first step in the solution of potential flow over single or multi-element airfoils is to define elements that describe the airfoil surface. We select a certain number of points on the airfoil surface and connect those points with straight or curved lines which define the panels (fig. 33).

Next, we *(1)* represent the body surface and its wake by a distribution of sources, doublets, and/or vortices of unknown strength, *(2)* parameterize the singularity strength (i.e., represent it by a polynomial), *(3)* enforce an appropriate boundary condition for the velocity field at control points, and *(4)* solve the resulting system of linear algebraic equations in unknown singularity strength.

In order to be able to treat lifting bodies it is necessary to introduce circulation, the strength of which is determined by the Kutta condition. Using an analysis of the potential flow near a sharp convex edge (Moran [48]), there are two possibilities: *(a)* the velocity at the trailing edge is infinite; and *(b)* the flow leaves the sharp trailing edge of an airfoil section in a smooth fashion. From the requirement of finite velocity at the trailing edge, the second alternative gives the following practical applications of the Kutta conditions: *(1) the streamline that leaves a sharp trailing edge is an extension of the bisector of the trailing edge, (2) the flow speeds on the upper and lower surfaces near the trailing edge are equal at equal distance from the trailing edge, and (3) if the trailing edge is not cusped, the flow stagnates there.* The most commonly used Kutta condition in surface singularity methods is that airflow leaves the trailing edge smoothly. This Kutta condition can be obtained by equating the trailing velocities over the upper and lower surface elements adjacent to the trailing edge,

$$v_{t_1} + v_{t_N} = 0 \tag{76}$$

This form of Kutta condition results in zero loading on elements nearest the trailing edge. In order to minimize the error due to this, we have to use very small elements in the trailing edge region. Other possibilities are to use an additional control point a finite distance from the trailing edge or to approximate the tangential velocity component at the trailing edge by two- or three-point quadratic extrapolation.

8 MATHEMATICAL DEVELOPMENT OF PANEL METHODS

Consider compressible, steady and irrotational flow of inviscid fluid in three dimensions. The fact that the flow is irrotational implies the existence of a velocity potential $\vec{\nabla}\Phi = \vec{V}$. The continuity equation then reduces to the full potential equation

$$(1 - \frac{\Phi_x^2}{a^2})\Phi_{xx} + (1 - \frac{\Phi_y^2}{a^2})\Phi_{yy} + (1 - \frac{\Phi_z^2}{a^2})\Phi_{zz} + \frac{2\Phi_x\Phi_y}{a^2}\Phi_{xy} + \frac{2\Phi_y\Phi_z}{a^2}\Phi_{yz} + \frac{2\Phi_z\Phi_x}{a^2}\Phi_{zx} = 0 \tag{77}$$

For slightly compressible flow we separate terms to

$$\Delta\Phi = f(\Phi) \tag{78}$$

Where f corresponds to the nonlinear terms. A simple Taylor iterative method then involves repetitive solution of the Poisson equations

$$\Delta\Phi^{(k)} = f(\Phi^{(k-1)}) \tag{79}$$

Further, for incompressible flow we have the familiar linear problem

$$\Delta\Phi = 0 \tag{80}$$

or in terms of stream function Ψ

$$\Delta\Psi = \frac{\partial^2\Psi}{\partial X^2} + \frac{\partial^2\Psi}{\partial Y^2} = 0 \tag{81}$$

Once Φ or Ψ is determined, the velocity field can be found as

$$u = \frac{\partial\Phi}{\partial X} = \frac{\partial\Psi}{\partial Y} \tag{82a}$$

$$v = \frac{\partial \Phi}{\partial Y} = -\frac{\partial \Psi}{\partial X} \tag{82b}$$

In order to solve a potential flow problem for incompressible flow, eq. (80) or (81) should be discretized with specified boundary conditions. We first convert the governing equation into an integral form. It is known that any incompressible flow can be represented by a distribution of sources and vortices over its boundary surface. To make the potential Φ single valued, we employ a branch cut to the far field (infinity). The body surface, outer boundary and cut surface are shown in fig. 34.

The associated boundary conditions are

$$\frac{\partial \Phi}{\partial n} = 0 \tag{83}$$

on the airfoil surface S_B and

$$\vec{\nabla}\Phi \rightarrow \vec{V}_\infty \ \ as \ \ r^2 = x^2 + y^2 \rightarrow \infty \tag{84}$$

in the far field. $\vec{V}_\infty$ is the specified uniform flow at infinity. On the outer boundary S_O, the boundary condition (84) can be approximated as

$$\frac{\partial \Phi}{\partial n} = \vec{V}_\infty \vec{n} \tag{85}$$

Where $\vec{n}$ is the unit vector outward and normal to S_B.

For lifting airfoil problems, the unknown circulation Γ is determined from the Kutta condition. For any simple closed contour C enclosing the airfoil the circulation Γ is defined as

$$\Gamma = \oint_C \vec{u}\, d\vec{s} = \oint_C \vec{\nabla}\Phi d\vec{s} = \Phi(U) - \Phi(W) \tag{86}$$

Where $\vec{u}$ is the velocity and U, W are a pair of adjacent points on either side of branch cut S_C. Across the cut we have

$$[\ \Phi\ ] = \Gamma \ \ on \ \ S_C \tag{87}$$

and the velocity is continuous, so

$$\left[\frac{\partial \Phi}{\partial n} \right] = 0 \quad on \quad S_C \tag{88}$$

Where [ ] denotes the jump across the branch cut. Now, let us introduce a source of unit strength at some point P in the integration domain A (i.e., flow field). In this case the fundamental solution of equation (80) satisfies

$$\Delta \chi = \delta(X - \xi, Y - \eta) \tag{89}$$

and is $\chi = \frac{1}{2\pi} ln r$; i.e., the potential of a source of unit strength. Where

$r = \sqrt{(X-\xi)^2 + (Y-\eta)^2}$

δ is the Dirac delta function, and

ξ, η are local coordinates of an arbitrary point in the integration domain A.

Multiplying eqs. (78) and (89) by χ and Φ, respectively, and subtracting, we get

$$\Phi \vec{\nabla}^2 \chi - \chi \vec{\nabla}^2 \Phi = \delta\Phi - \chi f \tag{90}$$

The divergence of the left hand side of eq. (90) is

$$\vec{\nabla}\Phi \ \vec{\nabla}\chi + \Phi \vec{\nabla} \ \vec{\nabla}\chi - \chi \vec{\nabla} \ \vec{\nabla}\Phi - \vec{\nabla}\chi \ \vec{\nabla}\Phi = \delta\Phi - \chi f \tag{91}$$

or

$$\vec{\nabla} \left(\Phi \vec{\nabla}\chi - \chi \vec{\nabla}\Phi \right) = \delta\Phi - \chi f \tag{92}$$

Integrating over the area A and applying the divergence theorem, we get

$$\int_A (\delta\Phi - \chi f) dA = - \int_S \vec{n} \left(\Phi \vec{\nabla}\chi - \chi \vec{\nabla}\Phi \right) dS \tag{93}$$

Integrating the right-hand side of eq. (93) by parts

$$\int_A \delta\Phi dA = \int_S (\chi \frac{\partial \Phi}{\partial n} - \phi \frac{\partial \chi}{\partial n})\, dS + \int_A \chi f dA \tag{94}$$

and evaluating the right side of eq. (94), we get the basic equation for panel formulation as follows

$$\beta\Phi(\xi,\eta) = \int_S (\chi \frac{\partial \Phi}{\partial n} - \Phi \frac{\partial \chi}{\partial n})\, dS \tag{95}$$

where the surface S consists of three components $S = S_B + S_C + S_O$

S_B is the body surface immersed in the flow

S_C is a two-sided surface on the cut

S_O is the surface of the outer boundary

and $\beta = \frac{\theta}{2\pi}$ (θ is the interior angle).

Hence, the governing equation becomes

$$\beta\Phi(\xi,\eta) = \int_{S_B+S_C+S_O} (\chi \frac{\partial \Phi}{\partial n} - \Phi \frac{\partial \chi}{\partial n})\, dS \tag{96}$$

Where, $\beta = 0$ if (ξ,η) is outside of the boundary, $\beta = \frac{1}{2}$ if (ξ,η) is on a smooth boundary and $\beta = 1$ if (ξ,η) is inside the boundary.

8.1 POTENTIAL BASED PANEL METHOD

- TOTAL POTENTIAL

If Φ in eq. (80) represents a total potential, then on the outer boundary S_O we have

$$\Phi = V_\infty X \cos\alpha + V_\infty Y \sin\alpha \tag{97}$$

where α is the angle of incidence of the flow.

Hence, the line integral becomes

$$\beta\Phi = V_\infty(X\cos\alpha + Y\sin\alpha) + \int_{S_B+S_C}(\chi\frac{\partial\Phi}{\partial n} - \Phi\frac{\partial\chi}{\partial n})\,dS \qquad (98)$$

Since $\partial\Phi/\partial n = 0$ on a solid surface, integrating by parts, we get

$$\beta\Phi = V_\infty(X\cos\alpha + Y\sin\alpha) + \int_{S_B}\Phi\frac{\partial\chi}{\partial n}dS - \int_{S_C}[\ \Phi\ ]\frac{\partial\chi}{\partial n}\,dS \qquad (99)$$

where $[\ \Phi\ ] = \Phi^+ - \Phi^-$ represents the potential jump across the wake that is equal to the circulation about the airfoil. In fact, $\frac{\partial\chi}{\partial n}$ may be considered as a doublet of unknown strength so that equation (99) represents a doublet panel integral equation that is important for the development of vortex panel methods used here in numerical experiments.

- PERTURBATION POTENTIAL METHOD

If Φ in eq. (80) represents the perturbation potential, then velocity can be written as

$$\vec{V} = \vec{V_\infty} + \vec{\nabla}\Phi \qquad (100)$$

In this case the potential vanishes on S_O, so that the line integral becomes

$$\beta\Phi = \int_{S_B+S_C}(\Phi\frac{\partial\chi}{\partial n} - \frac{\partial\Phi}{\partial n}\chi)\,dS = \int_{S_B+S_C}(\Phi\frac{\partial\chi}{\partial n} - \sigma\chi)\,dS \qquad (101)$$

Where $\sigma = \frac{\partial\Phi}{\partial n}$ represents source strength and is given as a boundary condition on the airfoil surface as

$$\frac{\partial\Phi}{\partial n} = -\vec{n}\,\vec{V}_\infty \qquad (102)$$

Equation (101) represents a source-doublet panel equation.

- VORTEX METHODS

A surface doublet distribution of density μ can be replaced by an equivalent surface vortex distribution. So, vortex based panel methods can be developed using eq.(94). Integrating by parts and setting $\gamma = \frac{\partial \Phi}{\partial S}, \int \frac{\partial \chi}{\partial n} dS = \theta$ and $\frac{\partial [\Phi]}{\partial n}|_{cut} = 0$ (since [Φ] is constant along the wake), we get

$$\beta\Phi(\xi,\eta) = V_\infty(X\,cos\alpha + Y\,sin\alpha) - \int_{S_B} \gamma\theta\, dS \tag{103}$$

or in terms of a normal velocity at each control point and with $\beta = \frac{1}{2}$, the vortex integral equation becomes

$$\vec{n}\,\vec{V}_\infty = -2\int_{S_B} \gamma S \frac{\partial \Phi}{\partial n} dS \tag{104}$$

Using the relation between doublet and vortex distributions we can develop the basic equation for source-vortex panel methods. Recalling the doublet and source equations (99, 101) and knowing the potential jump [Φ] = $\Phi^+ - \Phi^-$ and $\frac{\partial \Phi}{\partial n} = 0$ on the wake, we get

$$\beta\Phi = \int_{S_B} \Phi \frac{\partial \chi}{\partial n} dS + \int_{S_C} [\,\Phi\,] \frac{\partial \chi}{\partial n} dS - \int_{S_B} \frac{\partial \Phi}{\partial n} \chi dS \tag{105}$$

Integrating by parts,

$$\beta\Phi = -\int_{S_B} \gamma\theta ds - \int_{S_B} \sigma\chi dS \tag{106}$$

where $\sigma = \frac{\partial \Phi}{\partial n}$ is the unknown source strength and $\gamma = \frac{\partial \chi}{\partial n}$ the unknown vortex strength.

As a function of the normal velocity at each control point,

$$\vec{n}\,\vec{V}_\infty = -2\int_{S_B} \gamma \frac{\partial \theta}{\partial n} dS - 2\int_{S_C} \sigma \frac{\partial \chi}{\partial n} dS \tag{107}$$

To determine the appropriate boundary integral solution to (80) or (81) it is necessary to discretize the basic panel equations and introduce the Kutta condition as a constraint. A detailed discretization for the panel method with linear distribution of vorticity and a curved panel method is given in *appendices A and B*, respectively.

9 STREAM FUNCTION APPROACH

As there is no normal velocity at a solid surface, each solid surface is a streamline of the flow and on the corresponding streamline Ψ is a constant. Thus, for a multi-airfoil problem the boundary conditions for eq. (81) on an airfoil component k can be written as

$$\Psi = \Psi_k \tag{108}$$

For computation it is convenient to have all equations in a nondimensional form, where distances are dimensionless with respect to the chord length c, velocities with respect to the free stream velocity V_∞, and the stream functions with respect to the product $V_\infty\, c$. For a uniform flow incident to the positive x - axis at an angle of attack α the dimensionless form of the stream function becomes

$$\psi_k = y sin\,\alpha - x cos\,\alpha \tag{109}$$

When the airfoil surface is replaced by a vortex sheet, the sum of the stream function for a uniform stream and the stream function for the vortex sheet should be constant on the airfoil surface. This can be represented by the integral equation (fig. 35) as

$$\psi_k = y(S) cos\,\alpha - x(S) sin\,\alpha - \frac{1}{2\pi}\int_s \gamma(S') ln(S, S') ds' \tag{110}$$

where:

ψ_k is the unknown stream function value on k - th airfoil section

$\gamma(S')$ is the vorticity strength at arbitrary point S',

$r(S, S')$ is the distance between points S and S'

$x(S), y(S)$ *are coordinates of the point of interest* S

s, s' curvilinear coordinates measured along the airfoil surface starting at the trailing edge

To solve equation (110), the airfoil surface is divided into N small surface elements - panels, and the integral is approximated by a summation. Applying eq. (110) at a control point C_i we obtain

$$\psi_k + \sum_{j=1}^{N} \frac{1}{2\pi} \int_s \gamma(S_j) lnr(C_i, S_j) ds_j = y_{C_i} cos\,\alpha - x_{C_i} sin\,\alpha \qquad (111)$$

Assuming that we have N control points C_i, the problem of potential flow over an airfoil section is reduced to that of solving these N simultaneous equations.

The most immediately required result is the velocity distribution on the airfoil surface. Since the velocity inside the airfoil is equal zero, the discontinuity in tangential velocity across a vortex sheet is equal to the density of the vortex sheet. This implies that, in solving eq. (111) for $\gamma(S_j)$, we directly obtain the velocity distribution on the airfoil surface.

In application of this method, we assume that we have straight panels and constant (or linear) distribution of γ_j over each element. Then we get

$$\psi_k + \sum_{j=1}^{N} \gamma_j K_{ij} = RHS_i \;\; (i = 1, 2....N) \qquad (112)$$

where K_{ij} is the influence coefficient of the element j on control point i and RHS_i is the right hand side of equation (111) evaluated at control point i. Using the notation of fig. 36, the influence coefficients can be written

$$K_{ij} = \int_{s_j}^{s_{j+1}} ln[r(S_i, S_j)\, ds \qquad (113)$$

$$K_{ij} = \frac{1}{4\pi}[-b_1 ln(r_1) + b_2 ln(r_2)] - \frac{\Delta s}{2\pi} + \frac{b_3}{2\pi}[tan^{-1}\frac{b_2}{b_3} - tan^{-1}\frac{b_1}{b_3}] \quad for \ i \neq j \tag{114}$$

and

$$K_{ij} = \frac{\Delta s}{2\pi}[ln(\frac{\Delta s}{2} - 1] \quad for \ i = j \tag{115}$$

where

$$\Delta s = s_{j+1} - s_j$$

$$r_1 = (x_j - x_{C_i})^2 + (y_j - y_{C_i})^2$$

$$r_2 = (x_{j+1} - x_{C_i})^2 + (y_{j+1} - y_{C_i})^2$$

$$b_1 = \frac{1}{\Delta s}[(x_j - x_{C_i})(x_{j+1} - x_j) + (y_j - y_{C_i})(y_{j+1} - y_j)]$$

$$b_2 = \frac{1}{\Delta s}[(x_{j+1} - x_{C_i})(x_{j+1} - x_j) + (y_{j+1} - y_{C_i})(y_{j+1} - y_j)]$$

$$b_3 = \frac{1}{\Delta s}[(x_j - x_{C_i})(y_{j+1} - y_j) - (y_j - y_{C_i})(x_{j+1} - x_j)]$$

The vortex strength γ_j at the intersection of two panels is determined as

$$\gamma_j = \frac{\gamma_{j-1}(S_j - S_{j-1}) + \gamma_j(S_{j+1} - Sj)}{S_{j+1} - S_{j-1}} \tag{116}$$

where $j \neq 1$ and N.

It can be seen that K_{ij} and RHS_i are functions of the coordinates and the angle of attack. If we have M airfoil sections with N control points, the system of equations (112) becomes a set of $M + N$ equations for N unknowns γ_j and M unknowns ψ_k. M additional equations are determined using the Kutta condition for each airfoil section. For this purpose we use an additional control point at the extension of the bisector of the trailing edge and assume that the streamline through other control points of a

given section passes through this point, too. Applying eq. (112) to these additional point C_T, we can write the Kutta condition as follows

$$\psi_k + \sum_{j=1}^{N} \gamma_j K_{Tj} = RHS_T \tag{117}$$

Now, the complete system of equations for an airfoil section can be written in matrix form as

$$\begin{bmatrix} K_{1,1} & . & . & . & K_{1,N} & 1 \\ . & & & & . & . \\ . & & & & . & . \\ K_{N,1} & . & . & . & K_{N,N} & 1 \\ \multicolumn{6}{c}{Kutta\ condition} \end{bmatrix} \begin{bmatrix} \gamma_1 \\ . \\ . \\ \gamma_N \\ \psi \end{bmatrix} = \begin{bmatrix} RHS_1 \\ . \\ . \\ RHS_N \\ RHS_{N+1} \end{bmatrix} \tag{118}$$

Solution of the system (118) gives us a nondimensional vortex density γ_j and the stream function ψ. Now, the pressure coefficient at any point on the airfoil surface may be obtained from the velocities via the equation

$$C_p(x_j, y_j) = 1 - (\frac{v_{t_j}}{V_\infty})^2 = 1 - \gamma_j^2 \tag{119}$$

When this method is extended to multi-component airfoils, we have a different value of the stream function for each airfoil section. For example, if we have a common configuration slot - main airfoil - flap, the matrix

equation can be written as

$$\begin{bmatrix} K_{1,1} & \cdots & K_{1,3N} & 1 & 0 & 0 \\ \vdots & \cdots & \vdots & \vdots & \vdots & \vdots \\ K_{N,1} & \cdots & K_{N,3N} & 1 & 0 & 0 \\ K_{N+1,1} & \cdots & K_{N+1,3N} & 0 & 1 & 0 \\ \vdots & \cdots & \vdots & \vdots & \vdots & \vdots \\ K_{2N,1} & \cdots & K_{2N,3N} & 0 & 1 & 0 \\ K_{2N+1,1} & \cdots & K_{2N+1,3N} & 0 & 0 & 1 \\ \vdots & \cdots & \vdots & \vdots & \vdots & \vdots \\ K_{3N,1} & \cdots & K_{3N,3N} & 0 & 0 & 1 \\ Kutta\ condition\ for\ slot & & & & & \\ Kutta\ condition\ for\ main\ airfoil & & & & & \\ Kutta\ condition\ for\ flap & & & & & \end{bmatrix} \begin{bmatrix} \gamma_1 \\ \vdots \\ \gamma_N \\ \gamma_{N+1} \\ \vdots \\ \gamma_{2N} \\ \gamma_{2N+1} \\ \vdots \\ \gamma_{3N} \\ \psi_1 \\ \psi_2 \\ \psi_3 \end{bmatrix} = \begin{bmatrix} RHS_1 \\ \vdots \\ RHS_N \\ RHS_{N+1} \\ \vdots \\ RHS_{2N} \\ RHS_{2N+1} \\ \vdots \\ RHS_{3N} \\ RHS_{3N+1} \\ RHS_{3N+2} \\ RHS_{3N+3} \end{bmatrix} \tag{120}$$

It can be seen that only right hand sides of systems (118) and (120) include the angle of attack α. If we want to determine flow at different angles of attack, it is necessary to determine coefficient matrix K_{ij} once, and recalculate RHS_i as a function of the angle of attack.

9.1 A DESIGN METHOD

In analysis mode, values of coordinates x_i and y_i and influence coefficients K_{ij} are given by the airfoil geometry and systems of equations (118) and (120) solved for the surface velocity distribution γ_j. However, for the airfoil design problem either values of surface velocities γ_j or pressure distribution are given and the governing equations are to be solved for the airfoil geometry x_i and y_i. It is not possible to obtain a direct solution for geometry but instead we may apply an iterative procedure in which the geometry of a starting airfoil section is gradually modified until the desired velocity (pres-

sure) distribution is achieved. Substituting the desired values for stream function ψ^D and velocities γ_j^D in eqs. (112) and (117) while retaining the K_{ij} from the previous design iteration allows solution for the geometry of a modified section. During the iteration process, the x_i coordinates may be kept constant and new values of y_i calculated. For single airfoil design, the value of a stream function ψ is arbitrary since its effect is only to move the airfoil up or down relative to the $x - y$ coordinate system. However, in the case of multicomponent airfoils, the difference in stream functions between any two components determines the flow through the slot between the components, and specification of the stream function values depends on the type of design (i.e., given geometry of one or more sections - design of other components; or design of all components). To start the design process it is necessary to define an arbitrary basic airfoil shape to generate the initial set of influence coefficients K_{ij}. After that the influence coefficients for each new iteration are calculated using the coordinates obtained from previous iteration. For example, at iteration m of the design procedure for control points we have

$$y_i^m = \frac{1}{cos\,\alpha}[x_i sin\,\alpha + \psi_k^R + \sum_{j=1}^{N} K_{ij}^{m-1}\gamma^R] \;;\; i = 1,2...N \qquad (121)$$

and for the trailing edge points

$$y_T^m = \frac{1}{cos\alpha}[x_T sin\alpha + \psi_k^R + \sum_{j=1}^{N} K_{ij}^{m-1}\gamma^R] \qquad (122)$$

Since the location of the trailing point is very close to the trailing edge we may assume that y_T is the location of the actual trailing edge. In this way after each iteration we have the location of one point that determines the remaining points on the airfoil surface. The iteration process ends if either

condition

$$max_i[|y_i^m - y_i^{m-1}|] \leq \tau_1 \tag{123}$$

or

$$max_i[|C_{p_i}^m - C_{p_i}^{m-1}|] \leq \tau_2 \tag{124}$$

for tolerances τ_1, τ_2 is met. During the design process, the control points of the airfoils are adjusted along vertical lines x_i = const. This implies that we can not have adjustment of coordinates parallel to the freestream direction. In practical applications this limitation is relaxed because we have a prescribed chord length and we always can have an arbitrary x - coordinate distribution.

The coordinates y_i that are obtained from equations (121 and 122) are panel control points. The points on the airfoil surface can be determined by passing a curve through control points and interpolating or projecting a straight line through the control points using the trailing point as an actual trailing edge of an airfoil. In the latter method, however, a small error due to incorrect location of the trailing edge can propagate during iteration and produce a saw-shaped airfoil that can not satisfy the required pressure or velocity distribution. In order to avoid these difficulties, referring to fig. 37a, after each iteration we compute the actual position of the trailing edge using the first and last control points on the airfoil, x- coordinate of the trailing edge of the starting geometry and the additional trailing point. In this way we can compute the coefficient matrix K_{ij} more accurately and use the actual trailing edge as starting point for determination of the airfoil surface. For example, referring to fig. 37b, using the equation of a straight line through two points P_1 (trailing edge) and C_1 (the first control point) we can determine location of P_2 (the second point on the airfoil surface),

and so on.

10 DISCUSSION OF RESULTS

10.1 AIRFOIL ANALYSIS

For the analysis we selected three characteristic single airfoils: *cambered Karman -Trefftz, cambered Joukovski and a conventional NACA 2412 airfoil.* We also consider multi-component airfoils.

- EXAMPLE 1: Cambered Karman - Treftz airfoil at $\alpha = 5^o$ (fig. 38)

  For this test case we use the stream function approach with 41 panels, source vortex panel method and higher order (curved) panel method with linear distribution of vorticity with 60 panels. It can be seen that the source vortex method gives a larger pressure peak at the leading edge on lower surface in comparison with an exact solution obtained by conformal mapping.In fact, neither surface singularity method gives a completely accurate solution on the lower surface , but the curved panel method and stream function approach can handle leading and trailing edges approximately with the same order of accuracy.

- EXAMPLE 2: Cambered Joukovski airfoil at $\alpha = 2^o$ (fig. 39)

  For analysis of this airfoil we use the stream function approach with 41 panels and a higher order vortex panel method with 60 panels. It can be seen that both methods give very good agreement with exact solution. As in the first example, the cambered Joukovski airfoil gives different velocities on upper and lower surfaces (high loading near the trailing edge) that the Kutta condition (76) can not satisfy. Neither

source vortex nor linear vortex panel method applied here with this Kutta condition can handle the present airfoil with given number of panels, successfully.

- EXAMPLE 3: NACA 2412 airfoil at $\alpha = 8^o$ (fig. 40)

  This test case shows a comparison between linear vortex and stream function panel methods with 40 panels. It can be seen that both methods give approximately the same solution. The test case also proves that for airfoils with very small camber,trailing edge velocities are almost the same and the Kutta condition (76) can be applied.

- EXAMPLE 4: William's configuration at $\alpha = 0^o$ (fig. 41)

  For analysis of multicomponent airfoils the most popular test case is William's configuration with flap at 10^o or 30^o. In the present example we use the configuration with flap at 30^o. The problem is solved with 41 and 60 panels on each section.It can be seen that a better solution was obtained with 41 panels. The solution on the flap compares well with the exact solution, while on the lower surface of the main airfoil there is a small difference between panel and exact solutions. The same problem is observed by Seebohm and Newman [57] using a linear vortex panel method with quadratic extrapolation of the Kutta condition.The present method gives approximately the same form of solution on the main airfoil but gives a better solution on the flap.

- EXAMPLE 5: Arbitrary two component configuration

  Fig. 42 indicates an arbitrary two component configuration of two NACA 2412 airfoils. Flap chord is 30% of the main airfoil chord with

angle of deflection of 10°.The configuration is analysed at different angles of attack: 1°, 5° and 9°. The form of solution is similar to that obtained by Moran, Cole and Wahl [51].

10.2 AIRFOIL DESIGN

10.2.1 SINGLE AIRFOIL DESIGN

The iterative design procedure should converge on an airfoil design which gives exactly the required velocity distribution. For this purpose we apply velocity and coordinate convergence conditions

$$E_1 = |u_i^m - u_i^R| < \epsilon_1 \tag{125a}$$

$$E_2 = |y_i^m - y_i^{m-1}| < \epsilon_2 \tag{125b}$$

where superscript m indicates iteration level and R the required velocity distribution.

To start the design process it is necessary to specify a target velocity distribution. Convergence depends on how close a starting geometry is to the target geometry. If the starting geometry is far from the target design process will take a larger number of iterations.Two cases were tested.

- EXAMPLE 1: In this example we attempt to design William's main airfoil starting from a NACA 0012 airfoil (fig. 43).There is a big difference in trailing edges and also a big difference in pressure distribution in this region. Using the known velocity distribution, forty panels were used for discretizing the starting geometry. After 14 iterations the velocity convergence criterion ($\epsilon_1 = 0.001$) was satisfied and the solutions differ only slightly at the trailing edge.

- EXAMPLE 2: NACA 2412 (fig. 44) designed from a NACA 0012 at the same angle of attack ($\alpha = 8°$). The starting geometry was very close to the required geometry and after 9 iterations the coordinate convergence criterion was satisfied. The design process was very stable and very accurate.

10.2.2 MULTICOMPONENT AIRFOIL DESIGN

In multicomponent airfoil design one can consider: *(1) design of all components, (2) given main airfoil - design other components, and (3) given flap or slot - design of the main airfoil.* The second case is the most interesting and practical because of several requirements that should be satisfied in design of high lift devices. These requirements are: *(a) the flap and/or slot must retract into the main airfoil to form a good airfoil for cruising flight, (b) the deflection angles should provide reasonable increase of lift, and (c) the gap between main section and flap should provide enough energy to push the separation point of the boundary layer as much as possible towards the trailing edge of the flap.*

- EXAMPLE 1: Design of William's configuration with flap at 30° (fig. 45)

  The required velocity distribution was given from the exact solution at $\alpha = 0°$. The starting geometry was composed from a NACA 0012 as main section and from NACA 0009 as flap at zero angle of deflection. It is important that the initial flap angle and spacing are not critical because these parameters are quickly adjusted by the design process. Since the x - coordinates are not altered during the design process, the flap chord will change if the flap deflection is different from the

starting deflection. For example, if a particular final flap chord is desired, we can determine angle of deflection and set starting flap at that angle.

The required surface velocities on both components were supplied together with stream functions ψ_1 and ψ_2 on the main section and flap, respectively. Discretization employed 80 panels - 40 on each section. The design test case is very difficult because of a high velocity peak at the leading edges of both sections. Very small deviation in coordinates of the leading edge will cause a big change in pressure. It can be seen that pressure on the flap is very close to the required pressure distribution and that at the leading edge of the flap the pressure peak is a little overestimated. On the main section we have good agreement in pressure on the upper surface and at the leading and trailing edges, but underestimated pressure on the lower surface.

- EXAMPLE 2: Here we design the same configuration as in the first example but using the main William's airfoil and NACA 0009 as starting geometry (fig. 46). That is, we want to design a flap with a fixed main airfoil. During the design process the velocity distribution on the main airfoil is assumed to be unknown and is replaced by the values computed in the previous iteration, while geometry and location of the main section are held constant. It can be seen that after five iterations the pressure distribution on the flap is close to the required values but there is a big difference in pressure distribution on the main section. The reason is a relatively slow change of pressure distribution on the flap that can also be observed in fig. 42 for two NACA airfoils at different angles of attack. Final solution

was obtained after 9 iterations with good agreement with pressure distribution in analysis mode (fig. 41)

11 CONCLUSION

Surface singularity methods are a good means for solution of potential flow problems. They can handle both single and multicomponent airfoils in analysis and design processes. The stream function approach appears to be very stable and a simple panel method applicable for practical initial analysis and design airfoil problems.The method provides reduction in the number of panels in comparison with other surface singularity methods and the best results were obtained with 40 - 60 panels per airfoil section.The modified approach for computing the actual position of the trailing edge provides designed airfoils with smooth shapes and stable design process.

APPENDIX A

12 VORTEX PANEL METHOD WITH LINEAR VORTEX STRENGTH

The circulation density on each panel varies linearly from one corner to the other and is continuous across the corner

$$\gamma_{s_j} = \gamma_j + (\gamma_{j+1} - \gamma_j)\frac{s_j}{L_j} \tag{126}$$

Recalling the basic equation for the vortex panel method, the velocity potential at the i -th control point is

$$\Phi(x_i, y_i) = V_\infty(x_i cos\alpha - y_i sin\alpha) - \sum_{j=1}^{N} \int_{s_j} \frac{\gamma(s_j)}{2\pi} tan^{-1}(\frac{y_{C_i} - y_{C_j}}{x_{C_i} - x_{Cj}})\, ds_j \tag{127}$$

Applying the boundary condition

$$\frac{\partial \Phi(x_{Ci}, y_{C_i})}{\partial n_i} = 0 \;\; i = 1, 2...N \tag{128}$$

and carrying out the differentiation and integration of equation (127), we get a matrix equation

$$\sum_{j+1}^{N} K_{ij}\gamma_j = b_i \tag{129}$$

If we enforce the Kutta condition explicitly, i.e., $\gamma_1 = \gamma_N = 0$, we get an overdetermined system that can be solved using the *least- squares method.* The idea is to find γ_j so as to minimize

$$E^2 = \sum_{i=1}^{N+1}(\sum_{j=1}^{N} K_{ij}\gamma_j - b_i)^2 \tag{130}$$

The minimum of (130) is

$$\frac{\partial E}{\partial \gamma_m} = 0 = 2\sum_{i=1}^{N+1}(\sum_{j+1}^{N} K_{ij}\gamma_j - b_i)K_{im} \tag{131}$$

or

$$\sum_{j=1}^{N}(\sum_{i=1}^{N+1} K_{i,j}K_{im})\gamma_i = \sum_{i=1}^{N+1} b_i K_{im} \quad m = 1,2...N \tag{132}$$

Now, (132) represents a determined system. However, if we use the Kutta condition in the form of eq. (76), (127) can be discretized as follows

$$\sum_{j=1}^{N}(K_{ij}^{(N_1)} + K_{ij}^{(N_2)})\,\gamma_j = sin(\theta_i - \alpha) \tag{133}$$

Where the coefficients are

$$K_{i,j}^{(N_1)} = \frac{1}{2}BF + AG - K_{ij}^{(N_2)} \tag{134}$$

$$K_{ij}^{(N_2)} = B + \frac{1}{2}\frac{IF}{L_j} - \frac{(AD+BE)G_j}{L_j} \tag{135}$$

and

$$A = sin\,(\theta_i - \theta_j)\,;\; B = cos\,(\theta_i - \theta_j)$$

$$C = (x_{C_i} - X_j)^2 + (y_{C_i} - Y_j)^2$$

$$D = (X_j - x_{C_i})cos\,\theta_j + (Y_j - y_{C_i})sin\,\theta_j$$

$$E = (x_{C_i} - X_j)sin\,\theta_j - (y_{C_i} - Y_j)cos\,\theta_j$$

$$F = ln\,[1 + \frac{1}{C}(L_j^2 + 2L_jD)]$$

$$G = Atan[\frac{L_jE}{(C + DL_j)}]$$

$$H = (x_{C_i} - X_j)sin\,(\theta_i - 2\theta_j) + (y_{C_i} - Y_j)sin\,(\theta_i - 2\theta_j)$$

For $i = j$ the coefficients become

$$K_{ij}^{(N_1)} = -1 \;\; and \;\; K_{ij}^{(N_2)} = 1 \tag{136}$$

The solution of the system (133) gives us the unknown circulation densities. The tangential component of velocity can be computed as follows

$$V_{T_i} = cos\,(\theta_i - \alpha) + \sum_{j=1}^{N}(K_{ij}^{(T_1)} + K_{ij}^{(T_2)}) \tag{137}$$

where the coefficients for $i \neq j$ are

$$K_{ij}^{(T_1)} = \frac{1}{2}AF - BG - K_{ij}^{(T_2)}$$

$$K_{ij}^{(T_2)} = A + \frac{1}{2}\frac{HF}{L_j} + \frac{(BD - AE)G}{L_j}$$

and for $i = j$

$$K_{ij}^{(T_1)} = K_{ij}^{(T_2)} = \frac{\pi}{2}$$

Then, the pressure coefficient can be computed as

$$C_{p_i} = 1 - V_{T_i}^2 \tag{138}$$

APPENDIX B

13 CURVED PANEL DISCRETIZATION

We use a base coordinate system for an element: Referring to fig. (47) the coordinate system is oriented and positioned as shown.

a) ξ or x coordinate is tangent to the curve; *b)* normal projections of the x - axis of the ends of the curve S lie equal distance Δ to the right and left ($l = 2\Delta$); *c)* η - axis is normal to the curve; *d)* general point R on the curve has coordinates (ξ, η); and *e)* the distance between some arbitrary point $P(x, y)$ and $R(\xi, \eta)$ is $r = \sqrt{(x-\xi)^2 + (y-\eta)^2}$

The boundary curve is defined as function of the ξ - coordinate, i.e., $\eta = \eta(\xi)$. In a neighborhood of the origin the curve and vorticity have power series approximations

$$\eta = a\xi^2 + b\xi^3 + ... \tag{139}$$

$$\gamma = \gamma^{(0)} + \gamma^{(1)}s + \gamma^{(2)}s^2 + ... \tag{140}$$

or as a function of panel curvature

$$s = \xi + \frac{2}{3}c^2\xi^3 \tag{141}$$

$$ds = (1 + 2c^2\xi^2)\, d\xi \tag{142}$$

Factor c depends on panel curvature and it is obtained by three point curve fitting using the averaged element slope

$$c_j = \frac{\frac{l_j+l_{j+1}}{l_j+l_{j-1}}(tan\,\theta_j - tan\,\theta_{j-1}) + \frac{l_j+l_{j-1}}{l_j+l_{j+1}}(tan\,\theta_{j+1} - tan\,\theta_j)}{(l_{j-1} + l_j) + (l_j + l_{j+1})} \tag{143}$$

Using eq. (143), a second order curved panel formulation can be written

$$\sum_{j+1}^{N-1}[-2\int_{-\frac{l_j}{2}}^{+\frac{l_j}{2}}(\frac{1}{2}-\frac{2}{3}\frac{\xi}{l_j}\frac{c_i^2}{l_j}\xi^3)(1+2c_j^2\xi^2)(\frac{\partial\theta}{\partial n})_{ij}d\xi_j]\,\gamma_j +$$
$$\sum_{j+1}^{N-1}[-2\int_{-\frac{l_j}{2}}^{+\frac{l_j}{2}}(\frac{1}{2}+\frac{\xi}{l_j}+\frac{2}{3}\frac{c_j^2}{l_j}\xi^3)(1+2c_j^2\xi^2)(\frac{\partial\theta}{\partial n})_{ij}d\xi_j]\,\gamma_{j+1}$$
$$= \vec{n}_i\,\vec{V}_\infty \qquad (144)$$

Using the Kutta condition $\gamma_1 = \gamma_N$, system (144) becomes

$$\sum_{j=2}^{N-1} K_{ij}\gamma_j = RHS_i \qquad (145)$$

where K_{ij} represents the left-hand side and RHS_i the right-hand side of system (144). The influence coefficients for normal velocity component can be expressed as

$$K_{ij}^{(N)} = K_{ij}^{(N_1)} + K_{ij}^{(N_2)} \qquad (146)$$

$$K_{ij}^{(N_1)} = \int_{-\frac{l_j}{2}}^{+\frac{l_j}{2}}(\frac{1}{2}+\frac{\xi}{l_j}+\frac{2}{3}\frac{c_j^2}{l_j}\xi^3)(1+2c_j^2\xi^2)(\frac{\partial\theta}{\partial n})_{ij}d\xi \qquad (147)$$

$$K_{ij}^{(N_2)} = \int_{-\frac{l_{j-1}}{2}}^{+\frac{l_{j-1}}{2}}(\frac{1}{2}+\frac{\xi}{l_{j-1}}+\frac{2}{3}\frac{c_{j-1}^2}{l_{j-1}}\xi^3)(1+2c_{j-1}^2\xi^2)(\frac{\partial\theta}{\partial n})_{ij-1}d\xi \qquad (148)$$

Where

$$(\frac{\partial\theta}{\partial n})_{ij} = \frac{(y_i - Y_j)\,sin\,\theta_i + (x_(i) - X_j)\,cos\,\theta_i}{(x_i - X_j)^2 + (y_i - Y_j)^2} \qquad (149)$$

$$X_j = x_j + \xi cos\,\theta_j - c_j\xi^2 sin\,\theta_j$$

$$Y_j = y_j + \xi sin\,\theta_j - c_j\xi^2 cos\,\theta_j$$

Coordinates X_j, Y_j are new control points obtained by parabolic curve approximation for each panel, and

$$K_{ij} = 1 \quad for \quad i = j$$

$$K_{ij} = -1 \quad for \quad i = j + 1$$

The influence coefficient for tangential velocity component can be written as

$$K_{ij}^{(T)} = K_{ij}^{(T_1)} + K_{ij}^{(T_2)} \tag{150}$$

$$K_{ij}^{(T_1)} = \int_{-\frac{l_j}{2}}^{+\frac{l_j}{2}} (\frac{1}{2} + \frac{\xi}{l_j} + \frac{2}{3}\frac{c_j^2}{l_j}\xi^3)(1 + 2c_j^2\xi^2)(\frac{\partial\theta}{\partial n})_{ij} d\xi \tag{151}$$

$$(\frac{\partial\theta}{\partial n})_{ij-1} = \frac{-(y_i - Y_j)\cos\theta_i + (x_(i) - X_j)\sin\theta_i}{(x_i - X_j)^2 + (y_i - Y_j)^2} \tag{152}$$

$$K_{ij}^{(T_2)} = \int_{-\frac{l_{j-1}}{2}}^{+\frac{l_{j-1}}{2}} (\frac{1}{2} + \frac{\xi}{l_{j-1}} + \frac{2}{3}\frac{c_{j-1}^2}{l_{j-1}}\xi^3)(1 + 2c_{j-1}^2\xi^2)(\frac{\partial\theta}{\partial n})_{ij-1} d\xi \tag{153}$$

and

$$K_{ij}^{(T_1)} = -\frac{\pi}{2} + \frac{c_j l_j}{2} - \frac{c_j^3 l_j^3}{12} \quad for \quad i = j \tag{154}$$

$$K_{ij}^{(T_2)} = -\frac{\pi}{2} + \frac{c_{j-1} l_{j-1}}{2} + \frac{c_{j-1}^3 l_{j-1}^3}{12} \quad for \quad i = j + 1 \tag{155}$$

NACA 0012

ELIPTIC SCHEME $\xi_{xx} + \eta_{xx} = P(\xi,\eta)$ $\xi_{yy} + \eta_{yy} = Q(\xi,\eta)$						PARABOLIC SCHEME
GRID SIZE	ERROR	ITER	R	ω	CPU time (sec)	CPU time (sec)
27 x 20	0.0001	78	5	1.81	11.722	0.083
27 x 20	0.0001	71	4	1.81	10.708	0.088
27 x 20	0.0001	68	3	1.80	10.264	0.090
27 x 20	0.0001	57	4	1.82	8.649	0.090
61 x 28	0.0001	99	3	1.81	48.561	0.288
61 x 28	0.001	479	2	1.81	239.259	0.283
TWO NACA 0012 AIRFOILS - FLAP ANGLE 25^0						
69 x 20	0.0001	275	4	1.80	105.702	0.340
69 x 20	0.001	116	5	1.81	48.301	0.342

NACA 0012 ** PARABOLIC SCHEME ONLY (R = 3)		
GRID SIZE	N° OF GRID POINTS	CPU time
61 x 28	1708	0.252
80 x 25	2000	0.288
80 x 40	3200	0.469
100 x 40	4000	0.571
120 x 80	9600	1.354
120 x 100	12000	1.686

Table 1 - Comparision of the parabolic and elliptic schemes

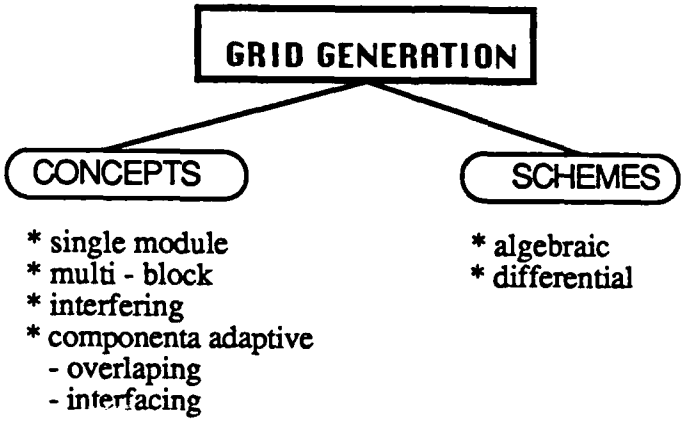


Fig. 1 Discretization methods

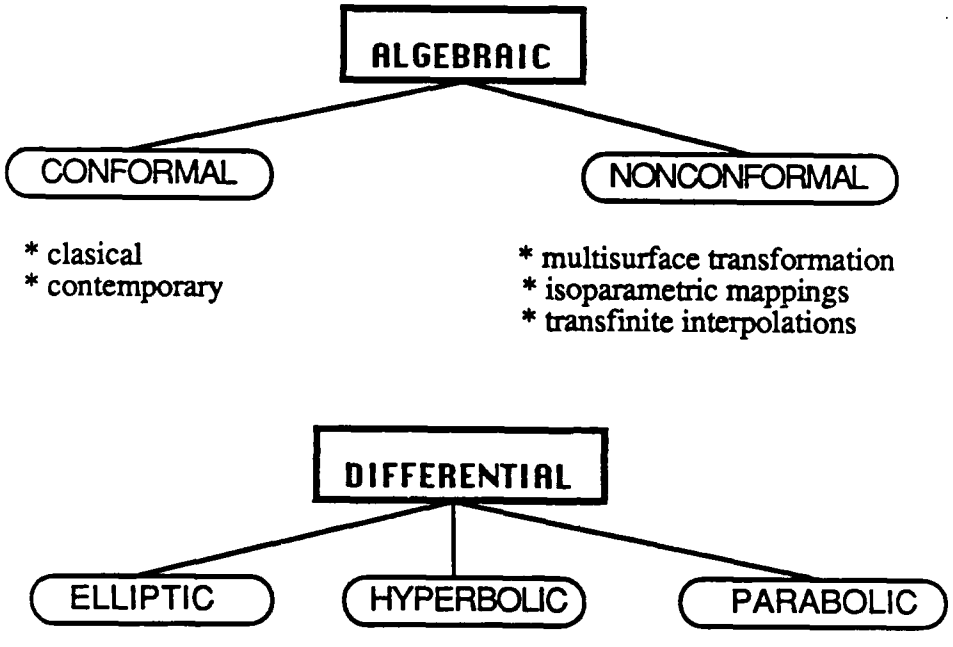


Fig. 2 Grid generation schemes

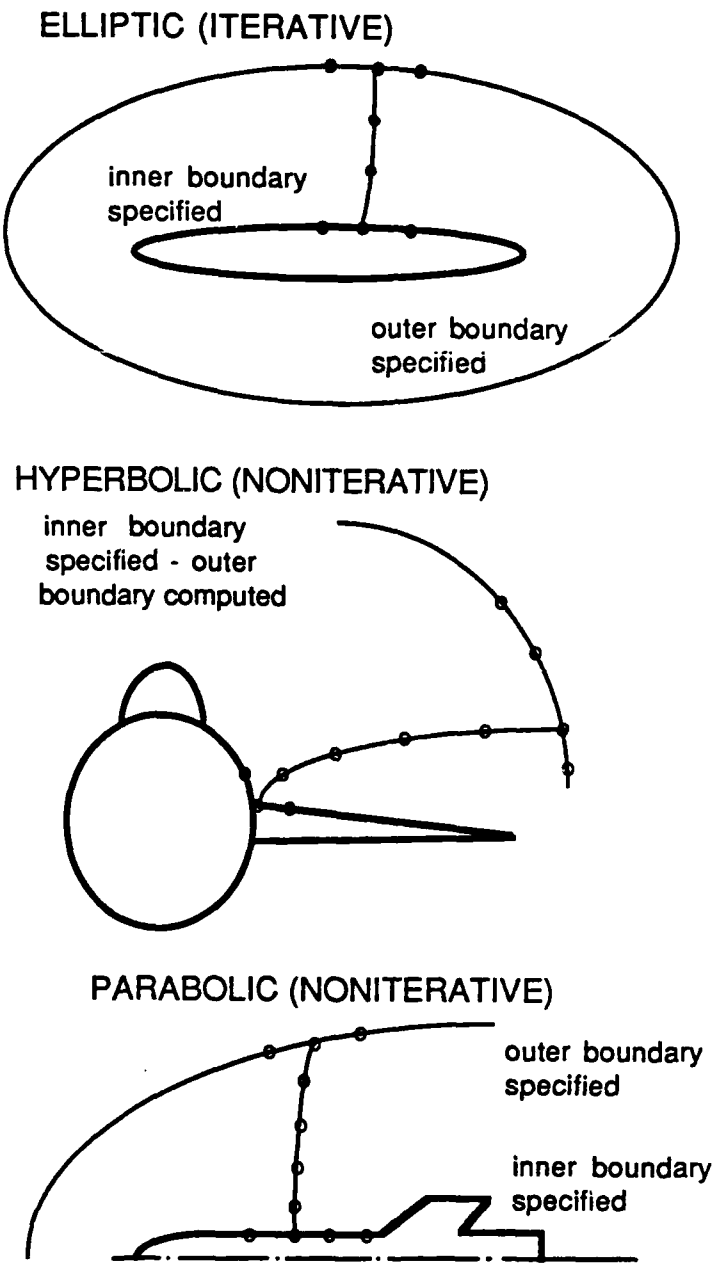


Fig. 3 Differential grid generation schemes

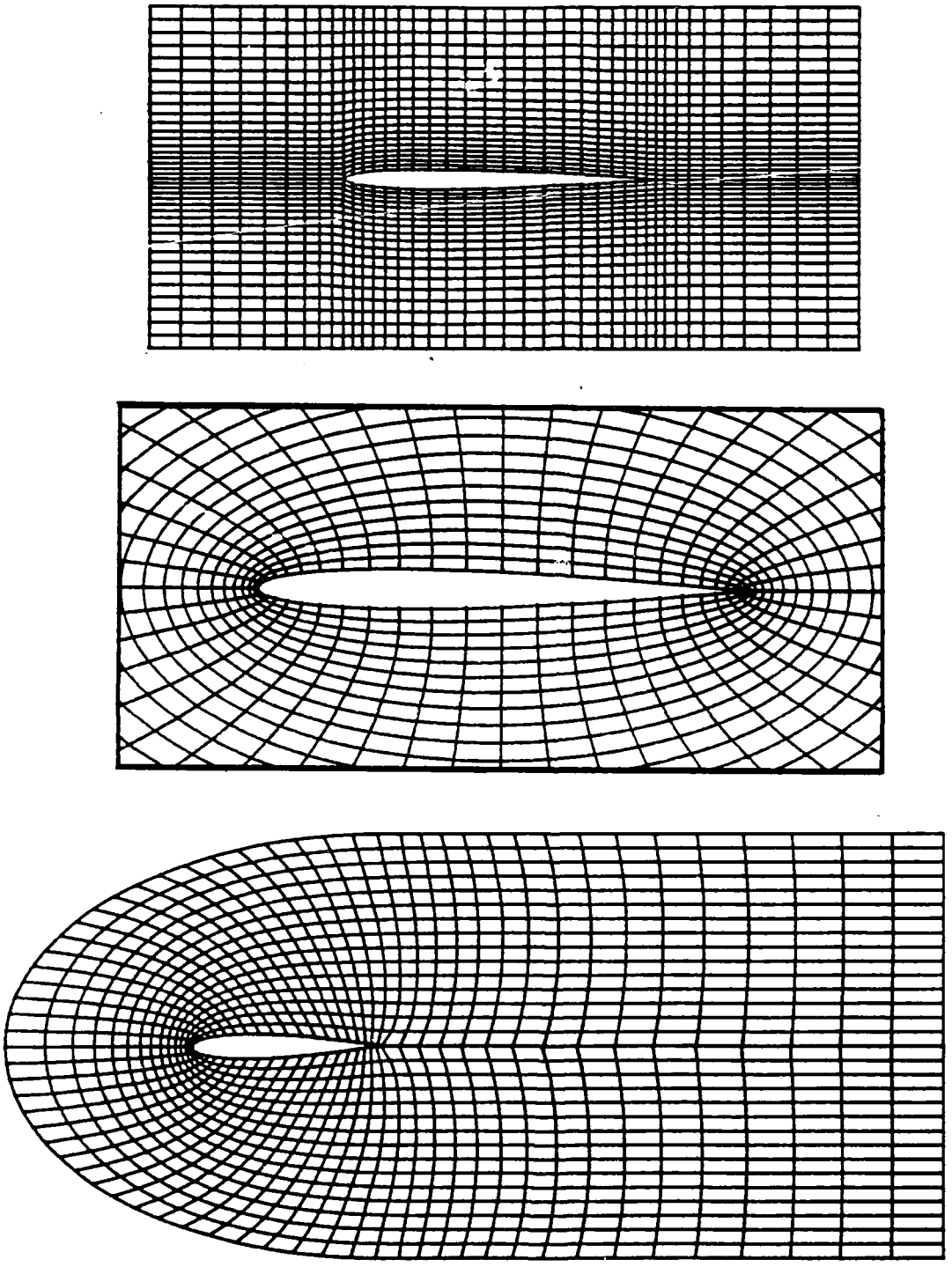


Fig. 4 Type of grids: a) a H-grid; b) an O-grid; c) a C-grid

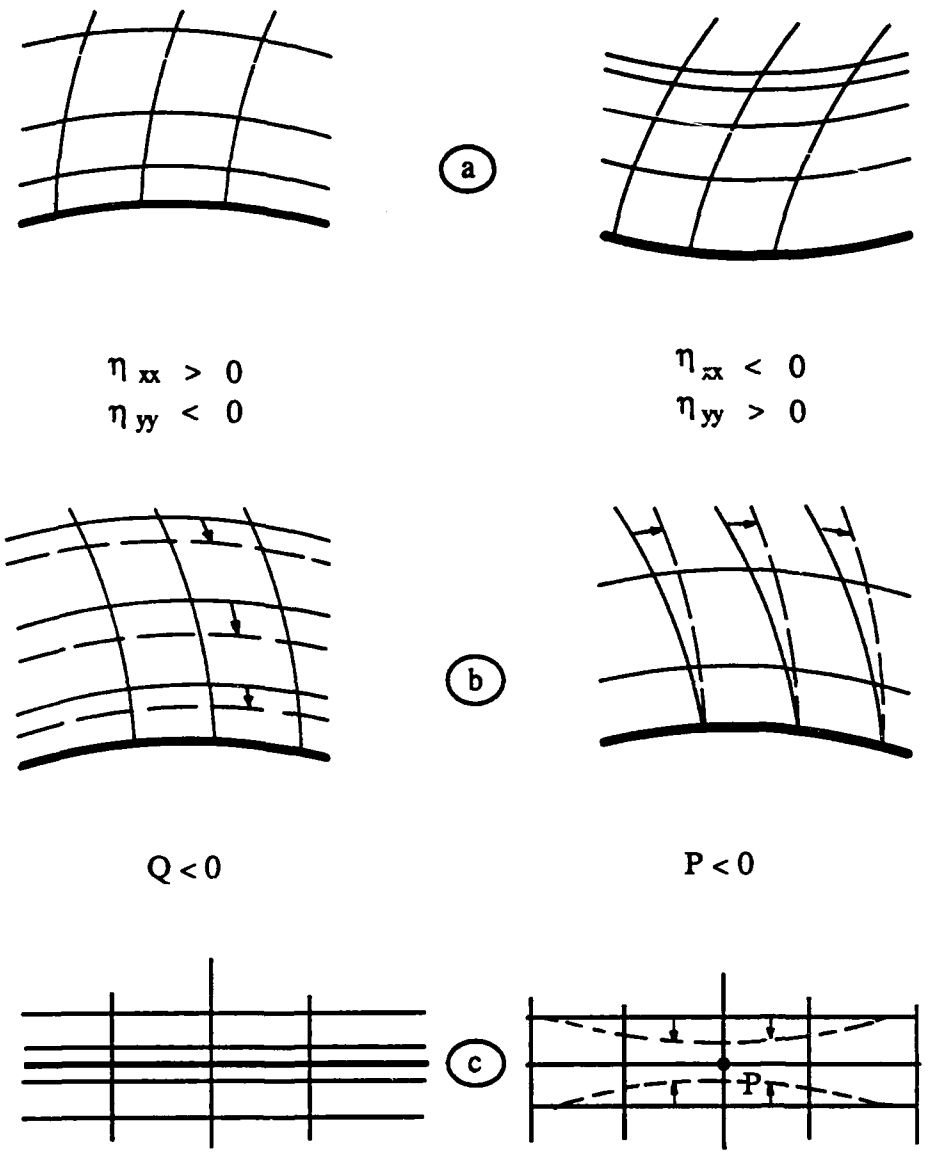


Fig. 5 Grid spacing control: a) The effect of the Laplacian; b) the effect of the forcing functions P and Q; c) the effect of the amplitude factors in the forcing functions

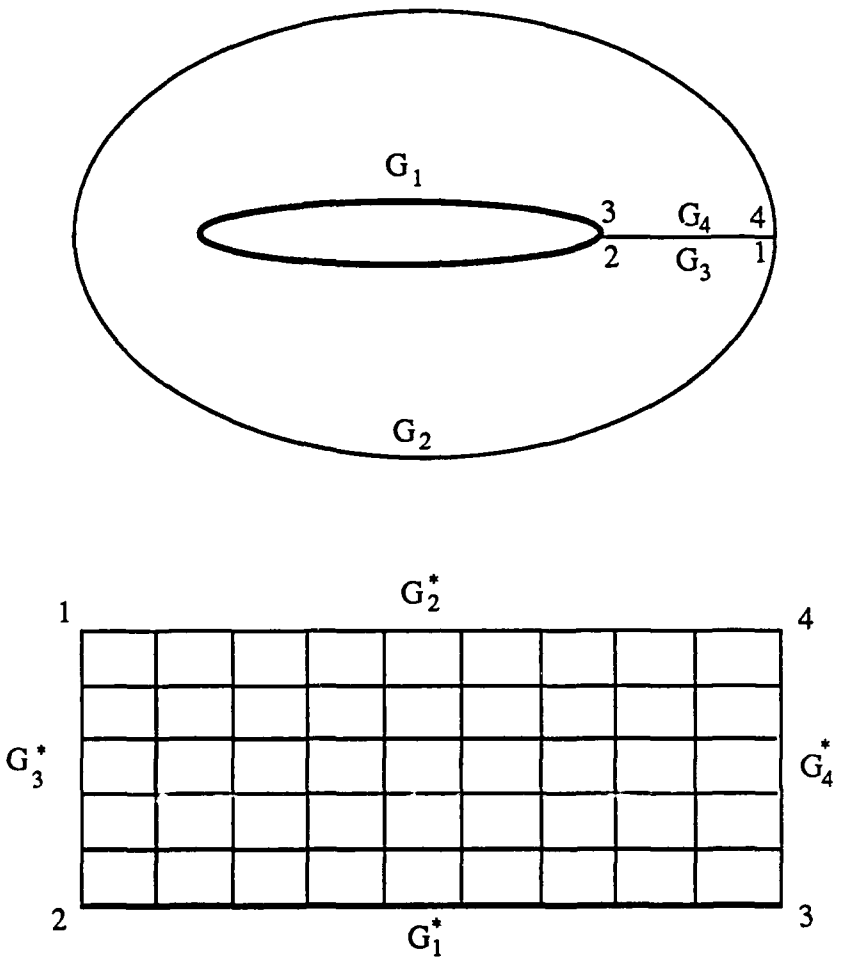


Fig. 6 Elliptic scheme:

Physical and Computational planes - Single body

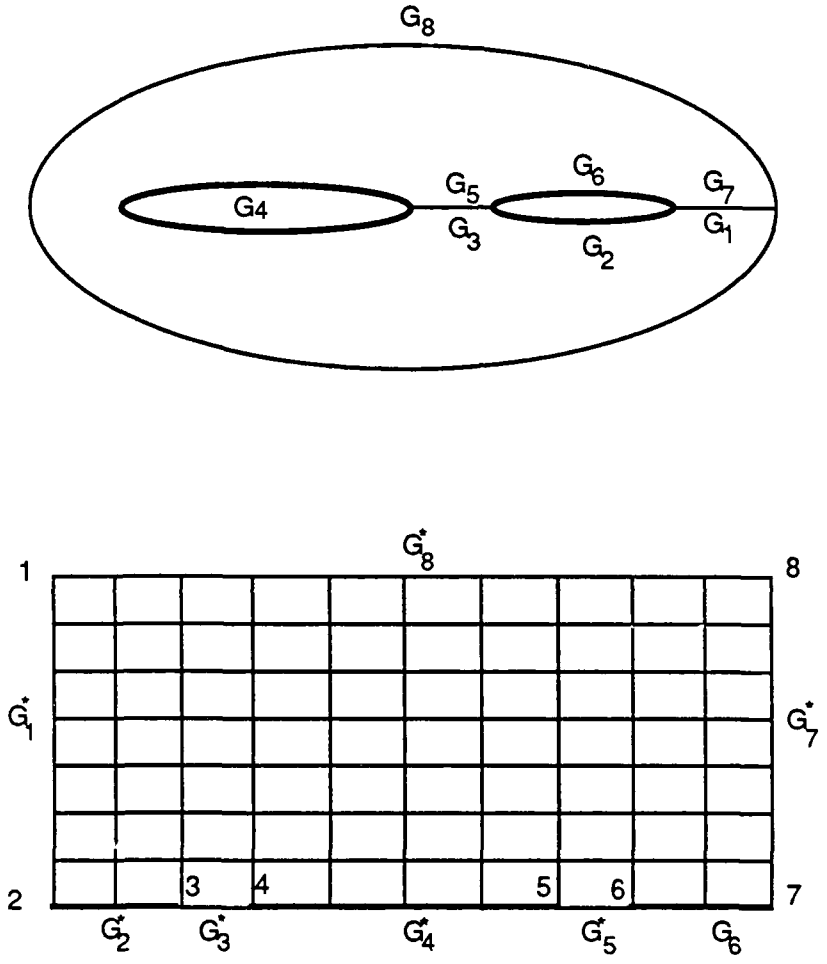


Fig. 7 Elliptic scheme:

Physical and Computational planes - Multiple bodies

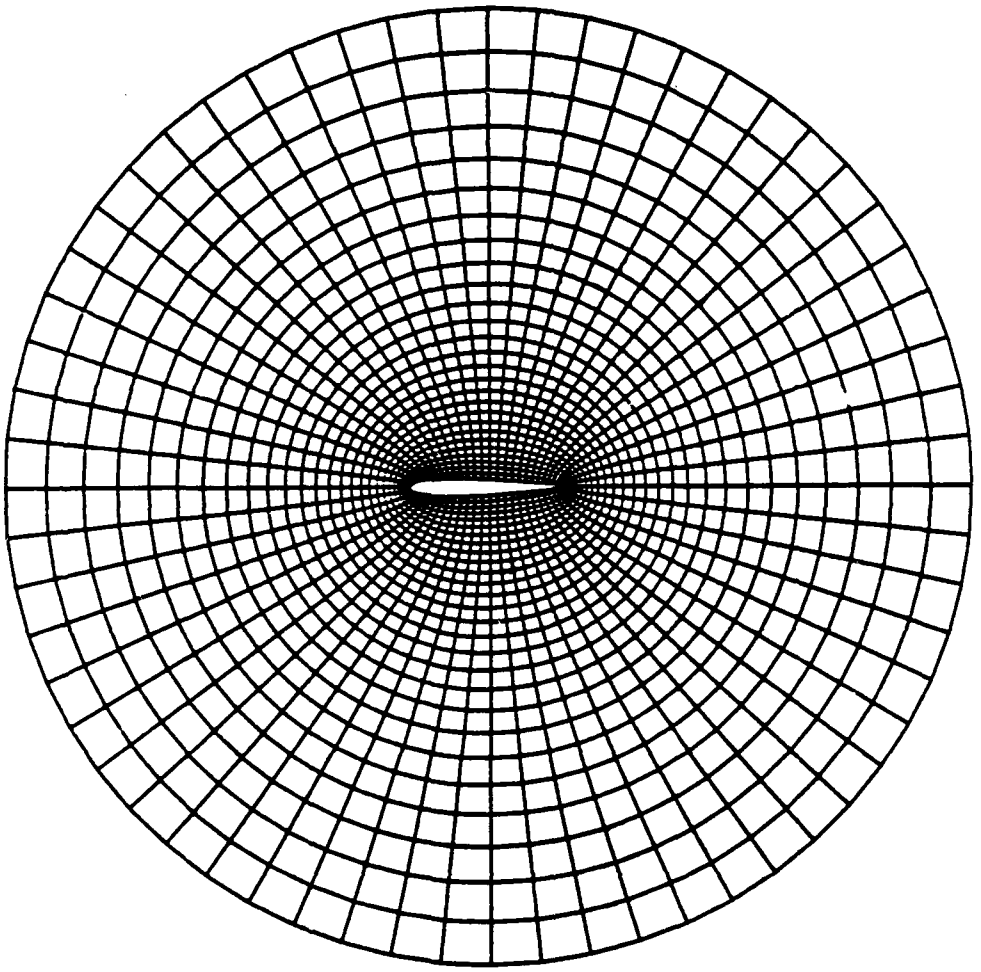


Fig. 8 O-grid about an NACA 0012 airfoil generated by the Poisson system (grid size 61 x 28, CPU time 48.561 sec)

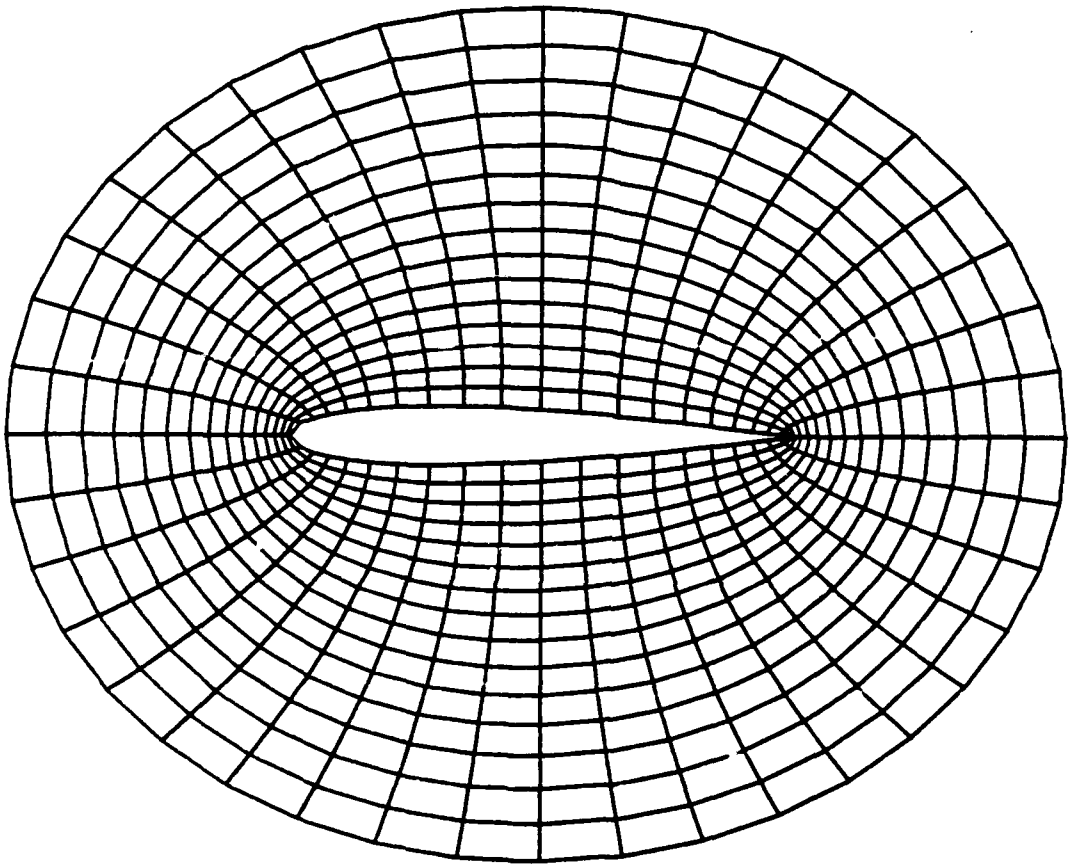


Fig. 9 O-grid about an NACA 0012 airfoil generated by the Laplace system (grid size 41 x 16, CPU time 1.971 sec)

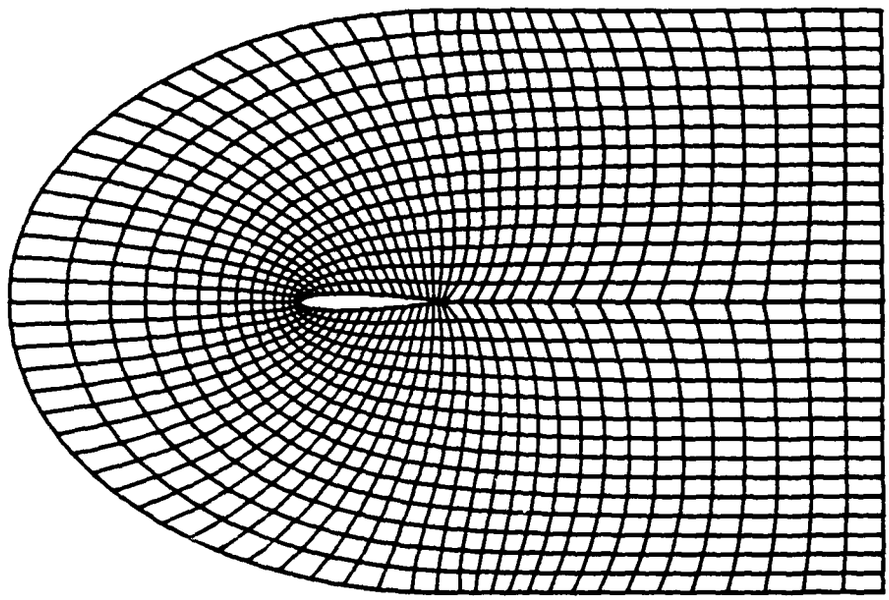


Fig. 10 C-grid about an NACA 0012 airfoil generated by the Laplace system

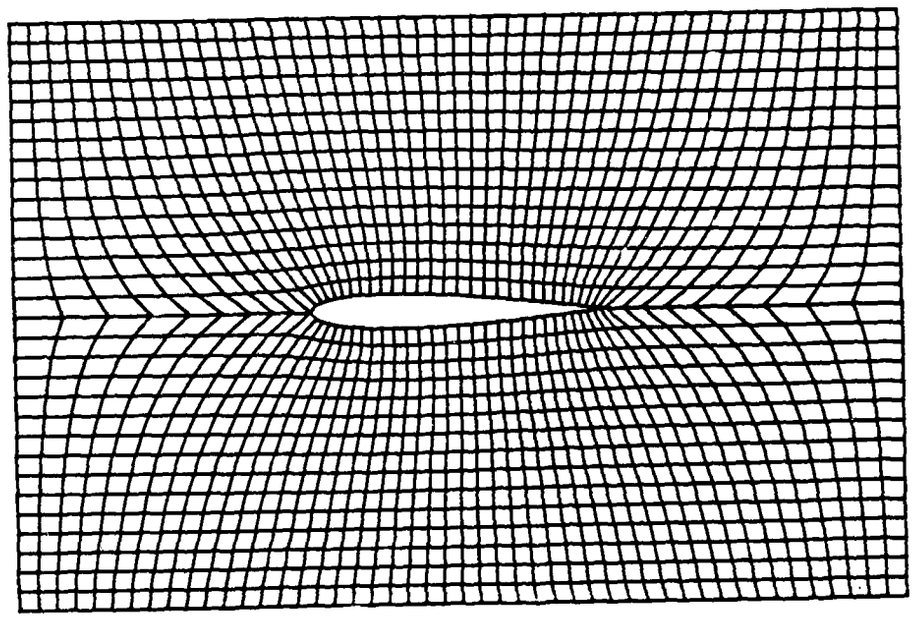


Fig. 11 H-grid about an NACA 0012 airfoil generated by the Laplace system

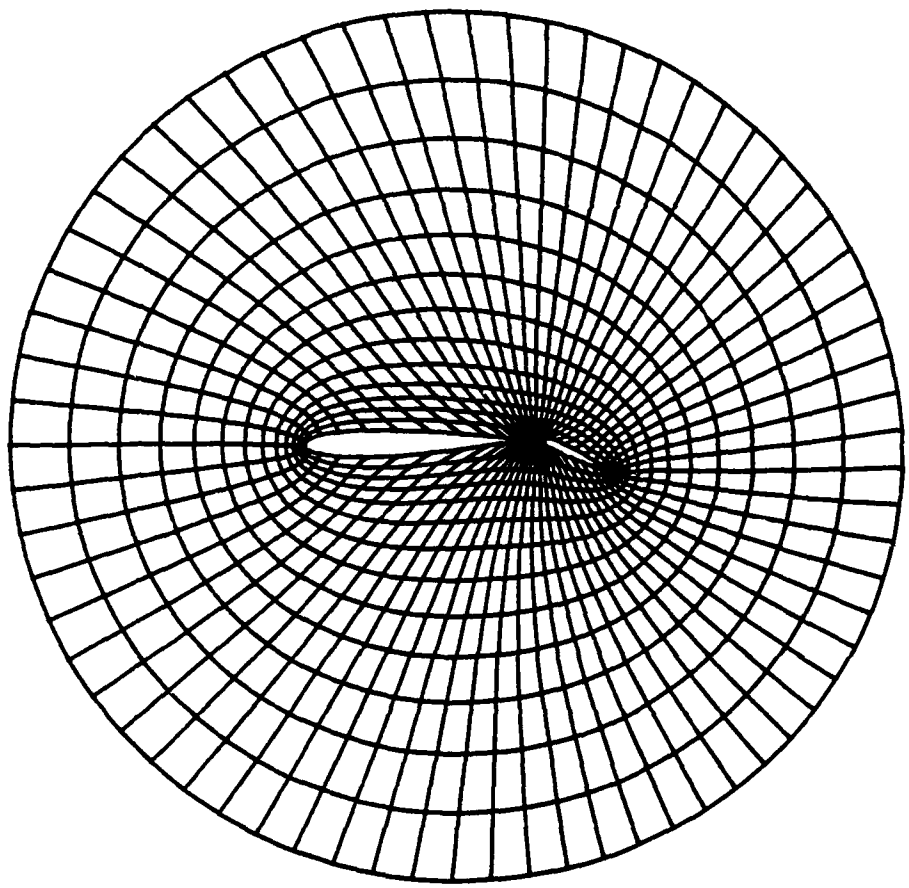


Fig.12 O-grid about two NACA 0012 airfoils generated by the Poisson system (flap angle 25°, grid size 69 x 20, CPU time 105.702 sec.)

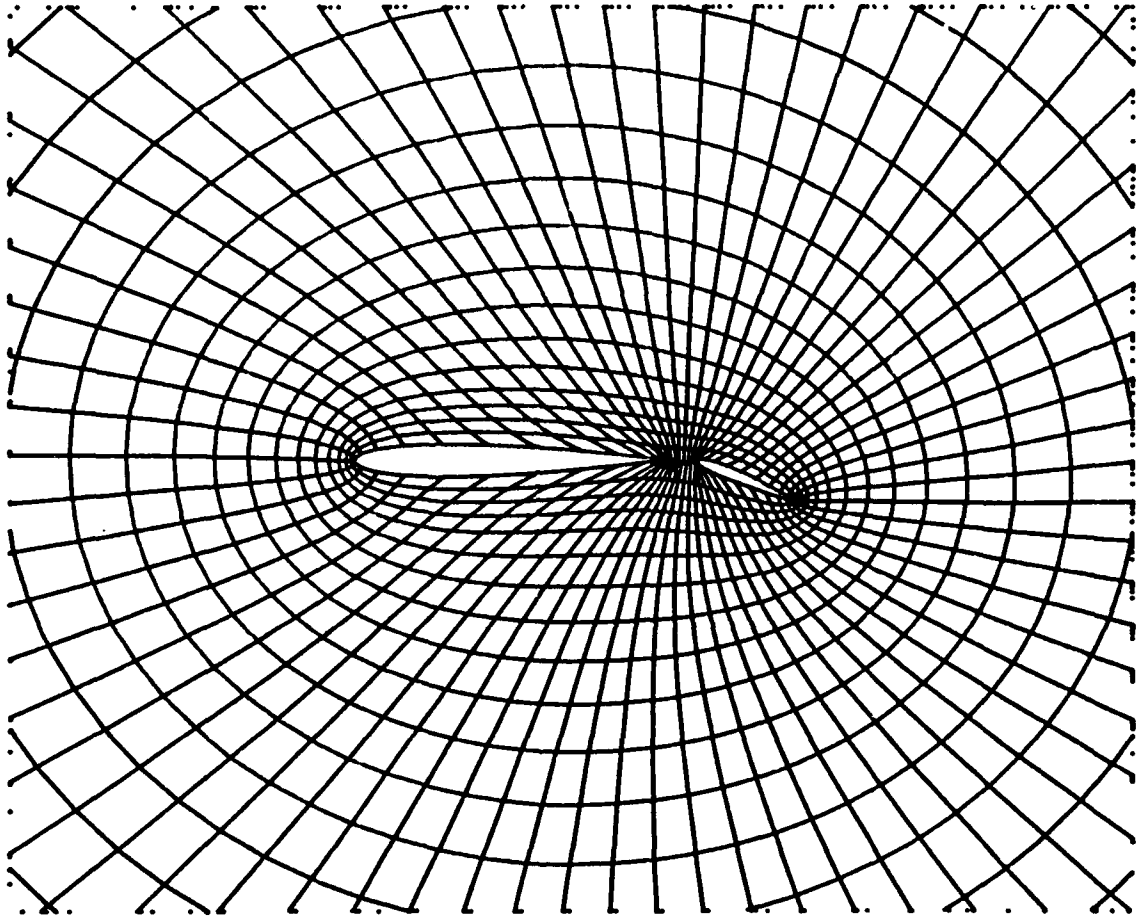


Fig. 13 Grid about two NACA 0012 airfoils - A detail

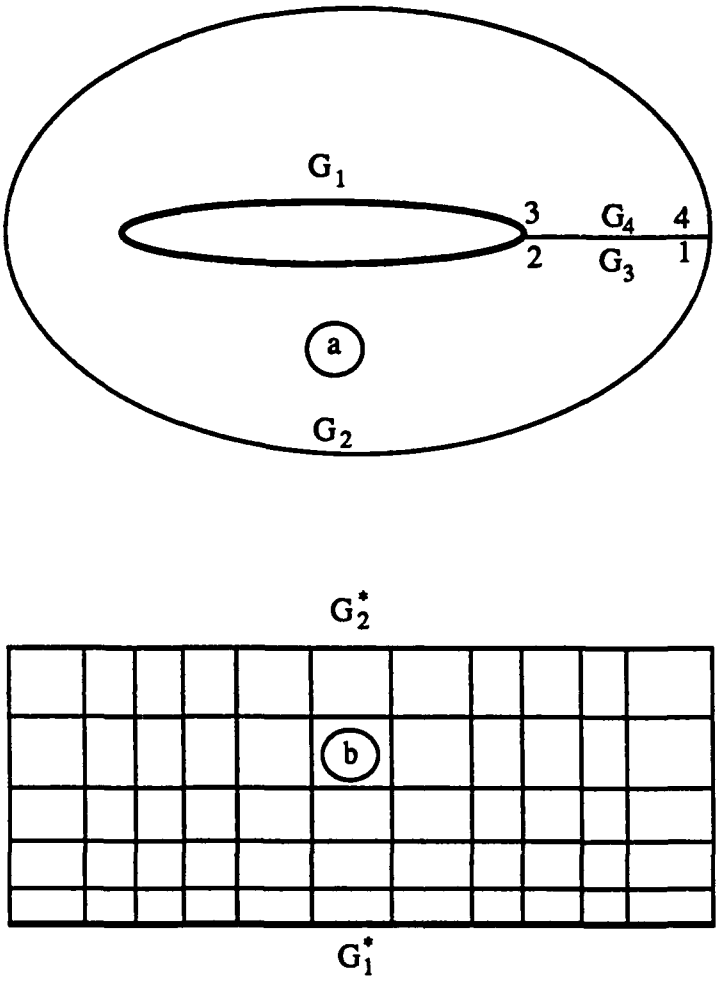
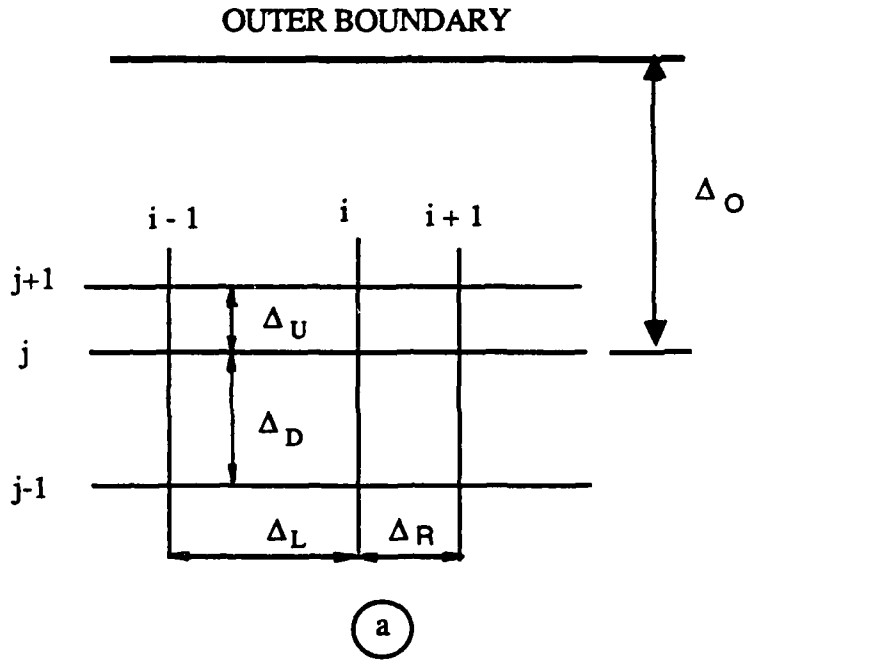


Fig. 14 Parabolic scheme:
a - Physical plane ; b - computational plane



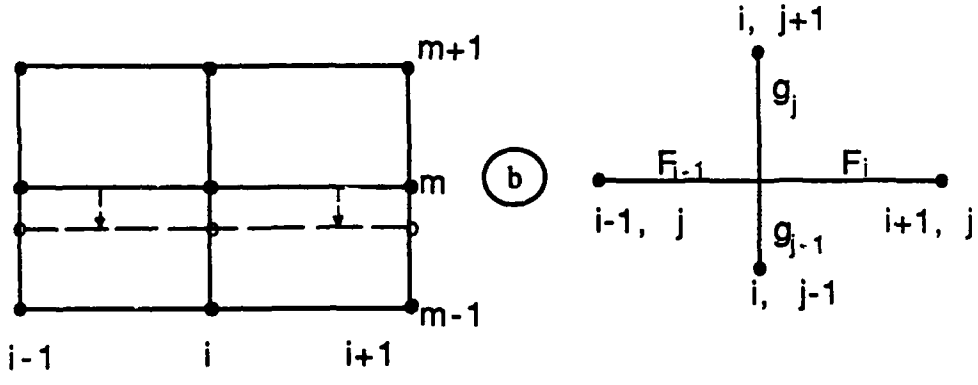


Fig.15 Definition of parameters for approximation of the model parabolic equations

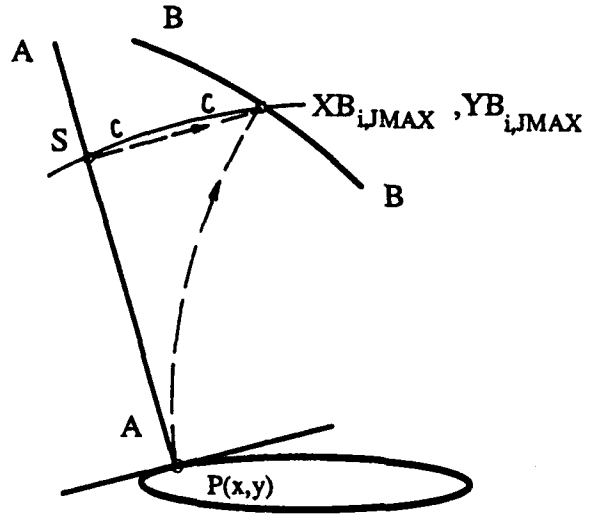


Fig. 16 Orthogonality control

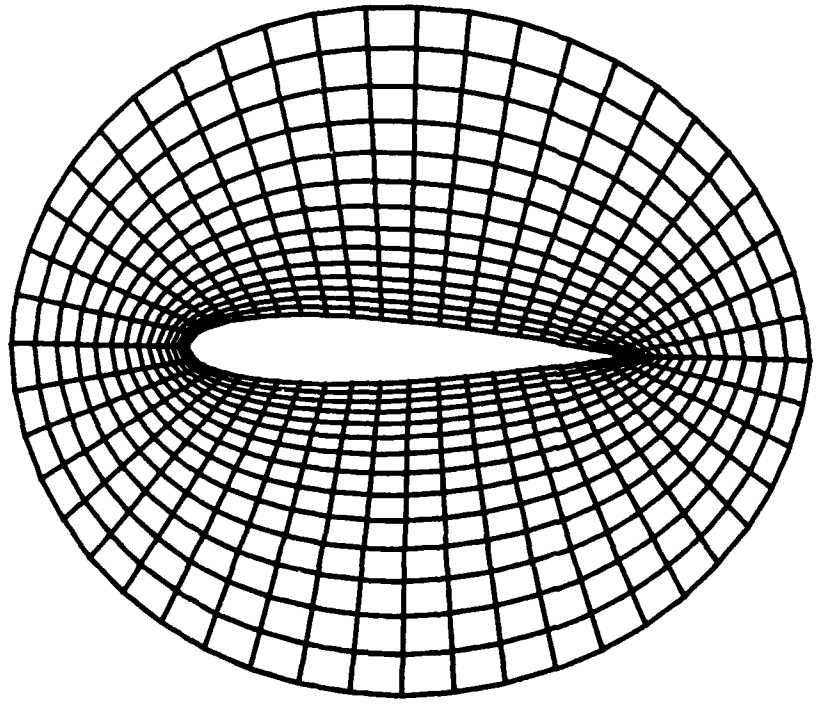


Fig. 17a CASE 2: A = A (ξ,η), B= 0, C=C (ξ,η)

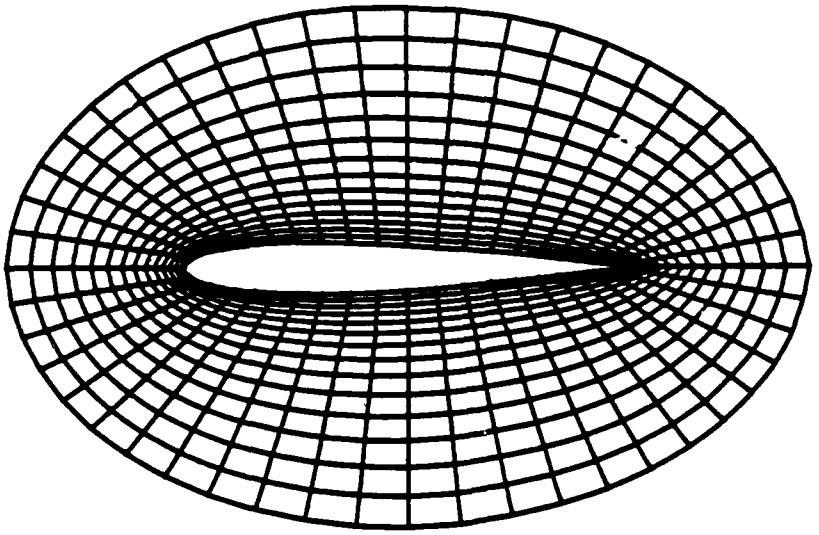


Fig. 17b CASE 3: A=A (ξ,η), B=0, C=1

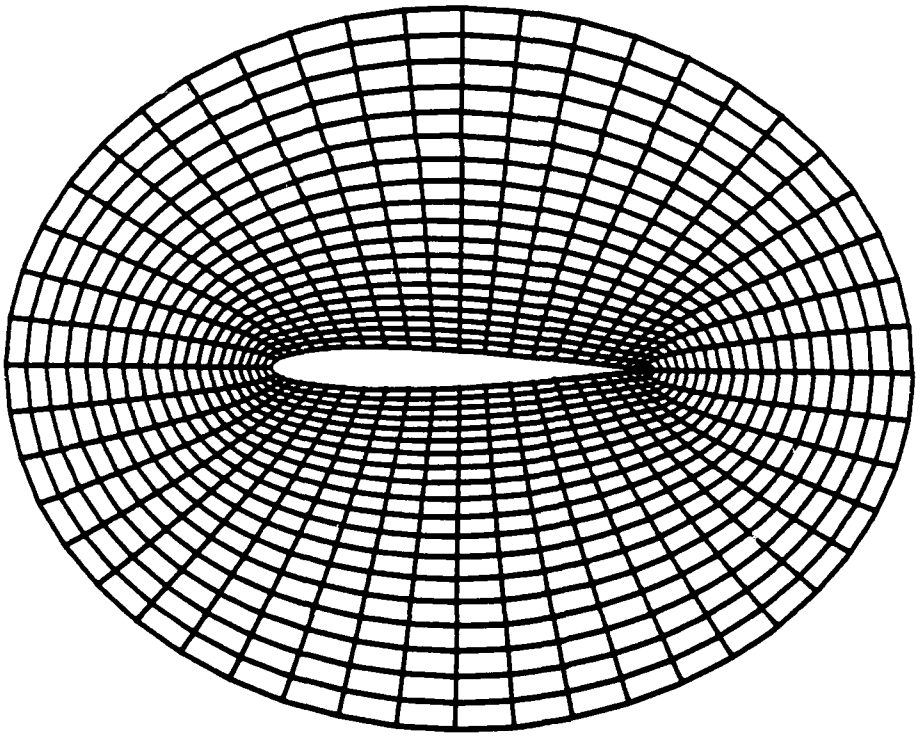


Fig. 17c CASE 4: A = 0, B=B(ξ,η), C=C(ξ,η)

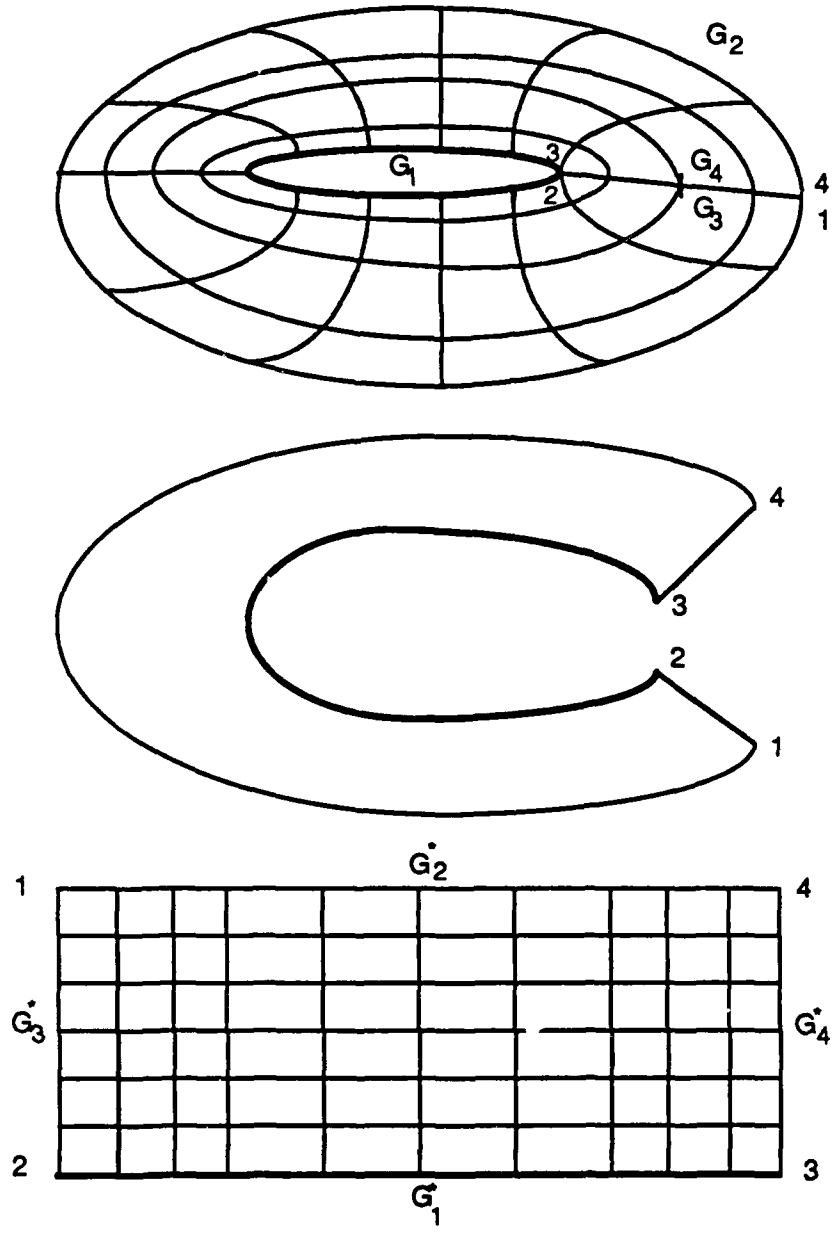


Fig. 18 An O-type grid forming

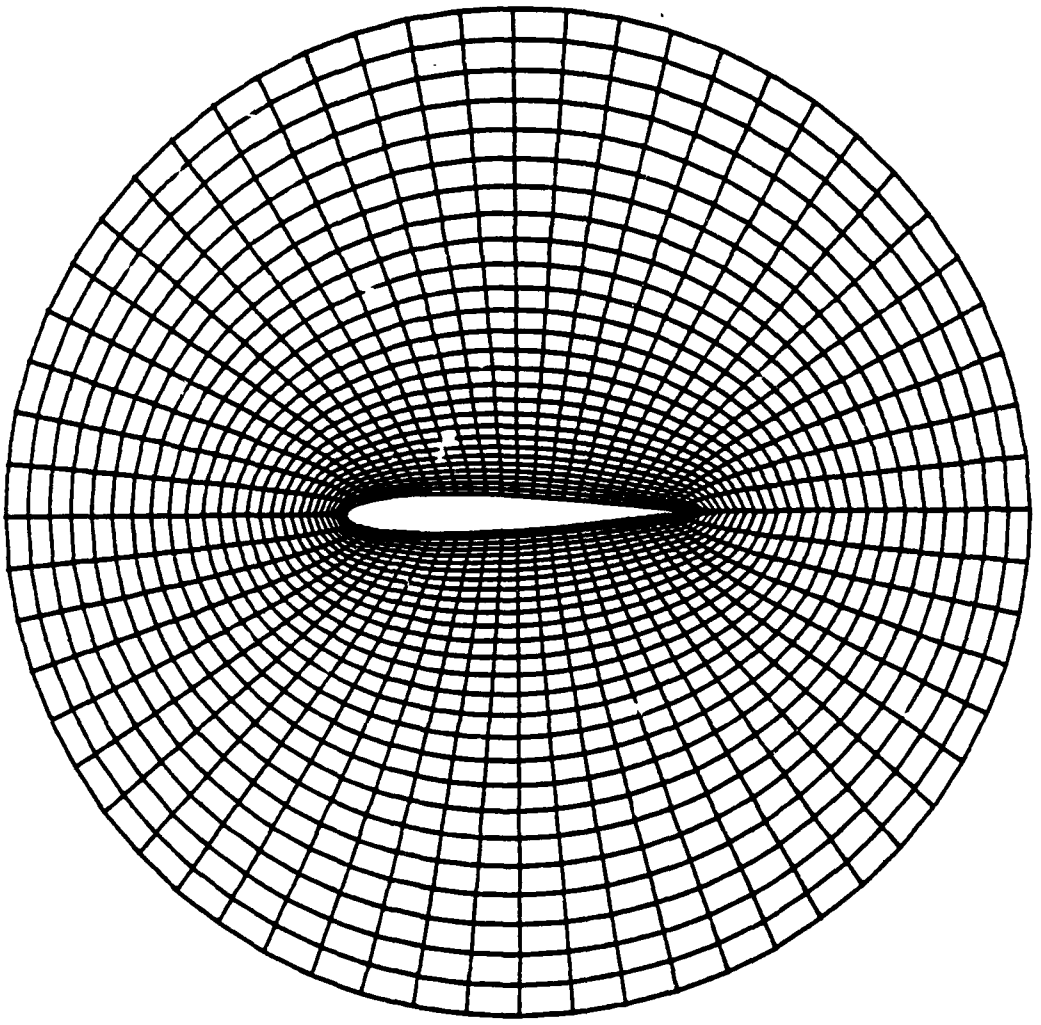


Fig.19 O-grid about an NACA 0012 airfoil generated by the Parabolic scheme (grid size 61 x 28, CPU time 0.258 sec)

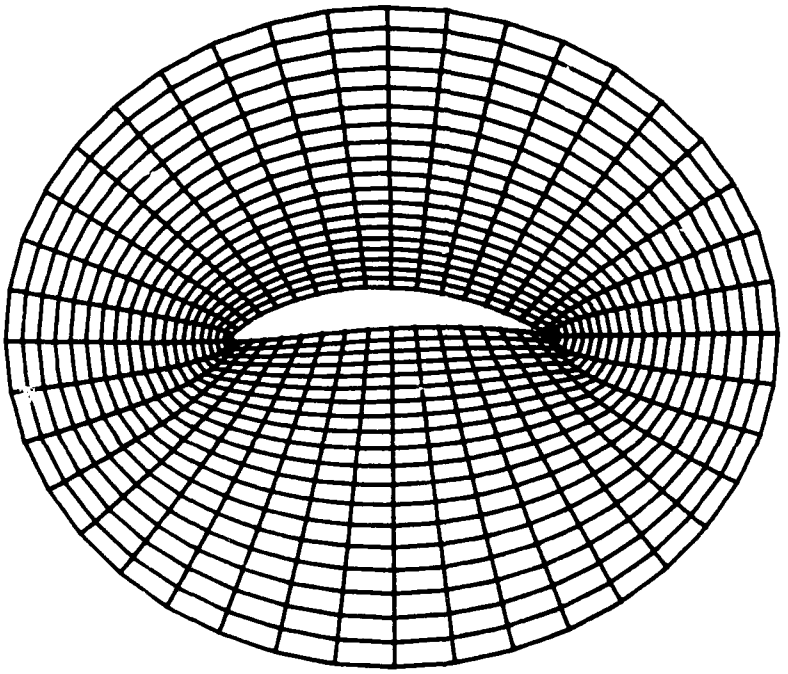


Fig.20 O-grid about a Karman-Trefftz airfoil generated by the Parabolic scheme (grid size 41 x 16, CPU time 0.189 sec)

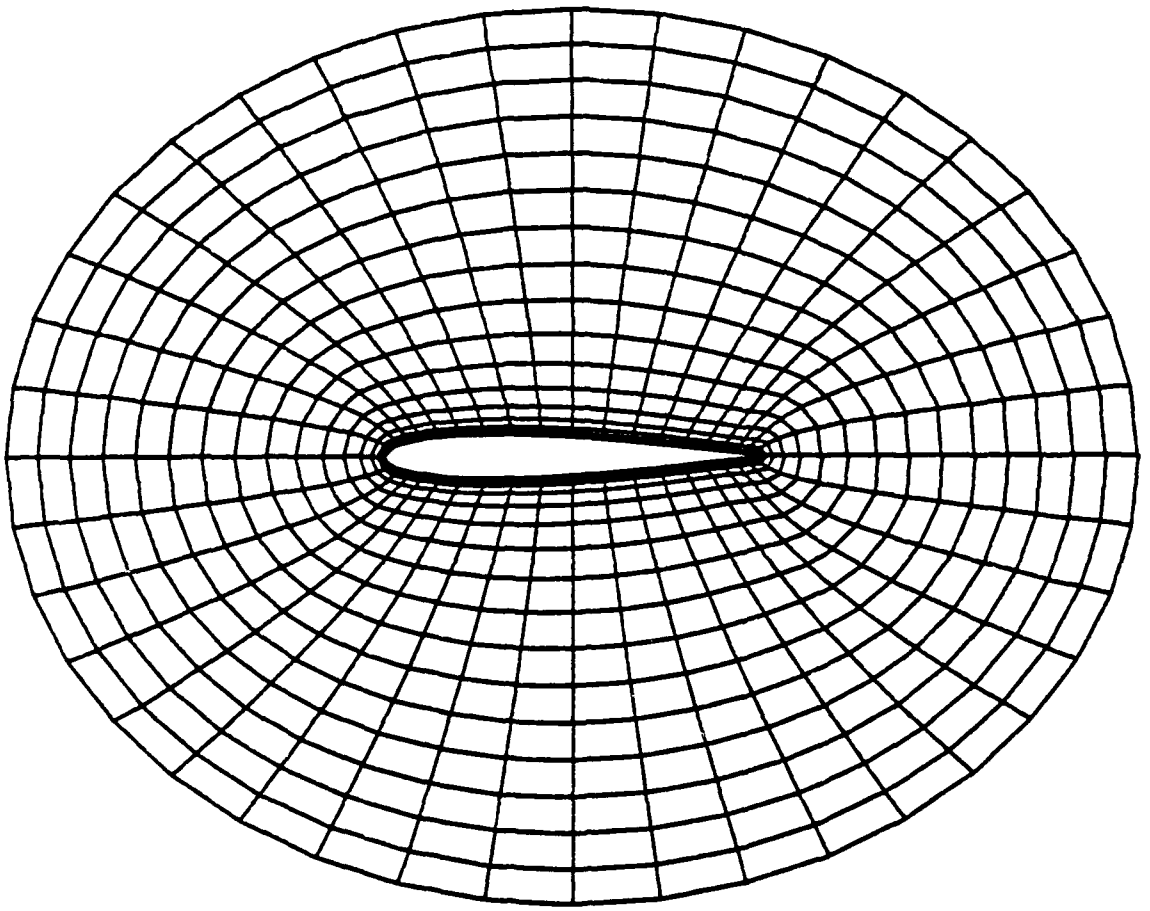


Fig.21 O-grid about an NACA 0012 airfoil generated by the Parabolic scheme with strong clustering at the airfoil surface
(grid size 41 x 20, CPU time 0.207 sec.)

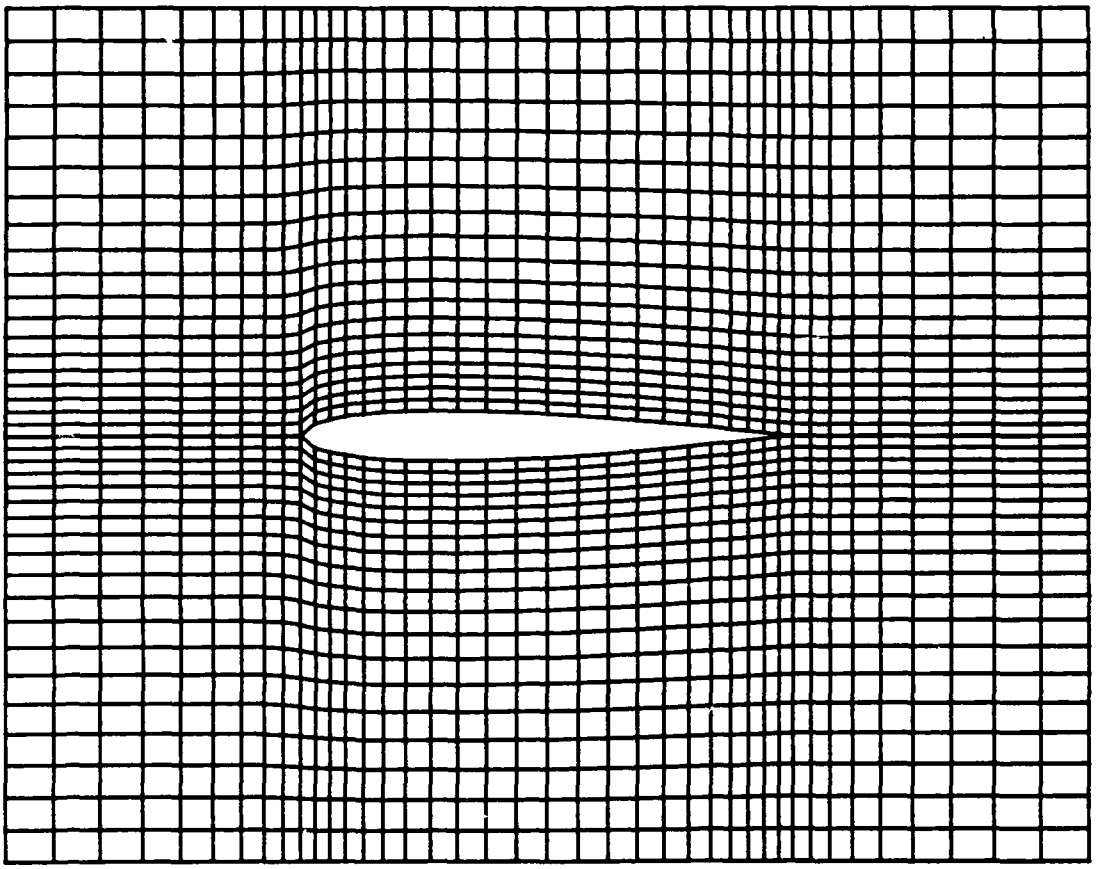


Fig.22 H-grid about an NACA 0012 airfoil generated by the Parabolic scheme without orthogonality at the airfoil surface
(grid size 41 x 20, CPU time 0.126 sec.)

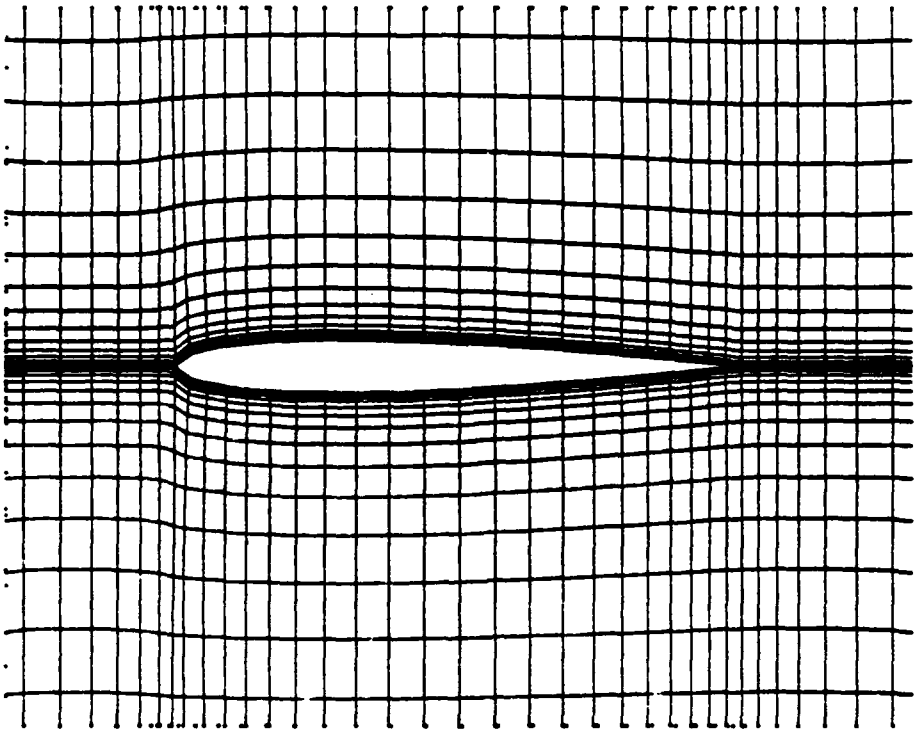


Fig. 23 H-grid generated by the parabolic scheme with strong clustering and orthogonality at the airfoil surface

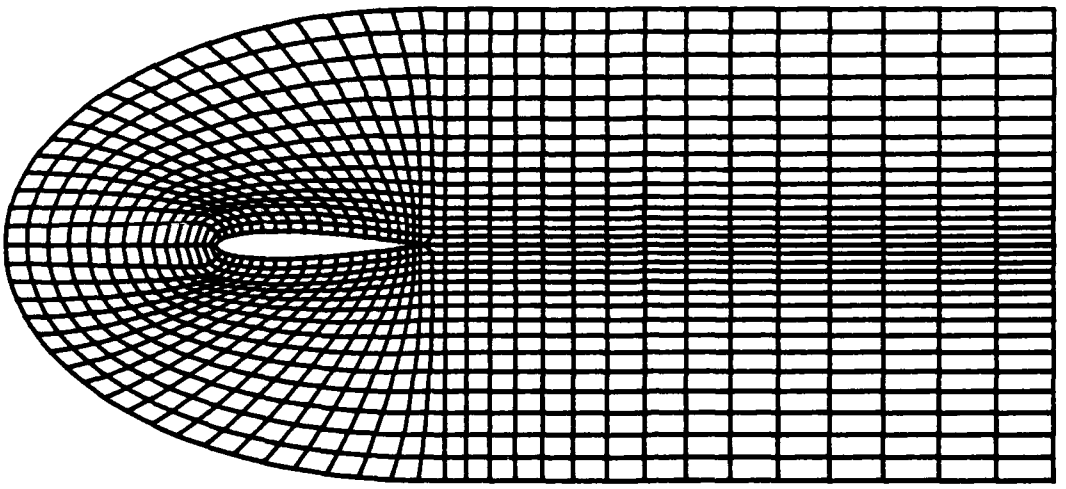


Fig. 24 C-grid about an NACA 0012 airfoil generated by the parabolic scheme

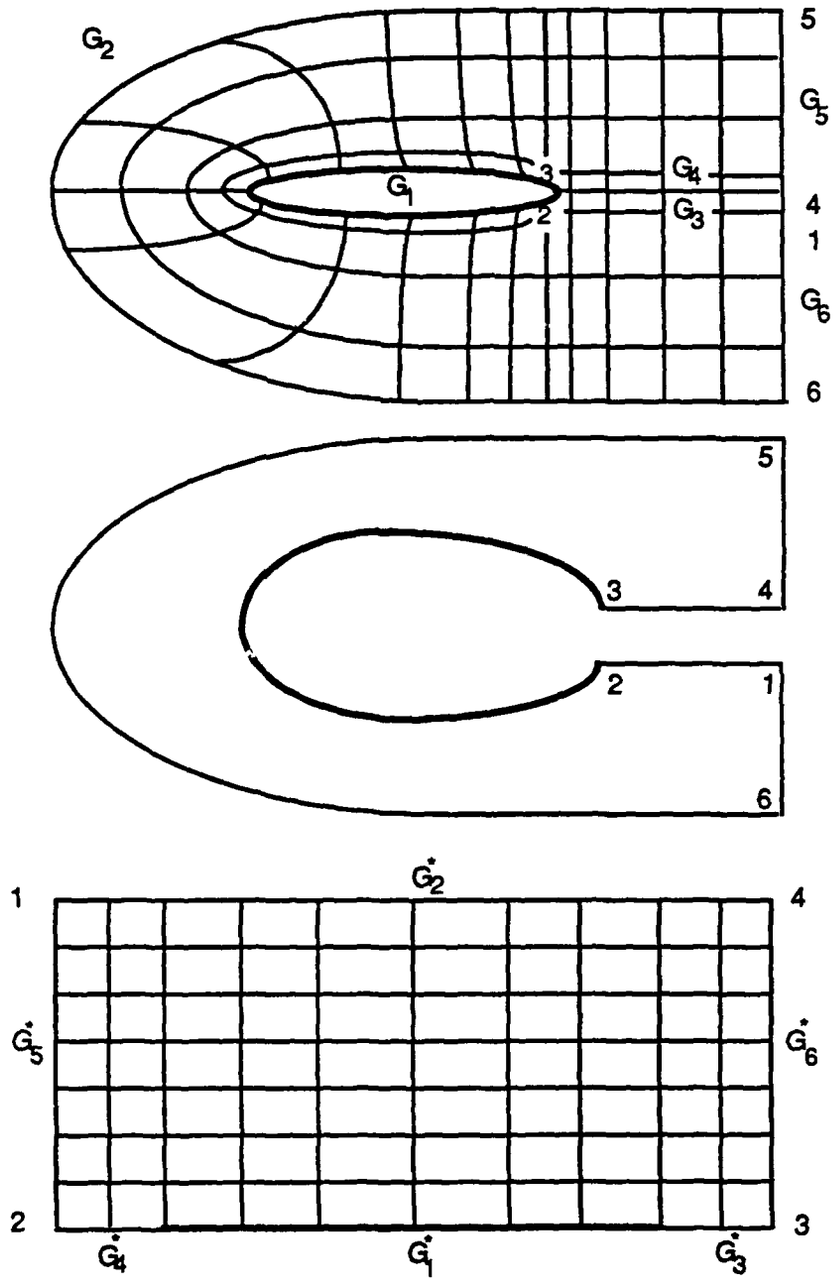


Fig. 25 A C-type grid forming

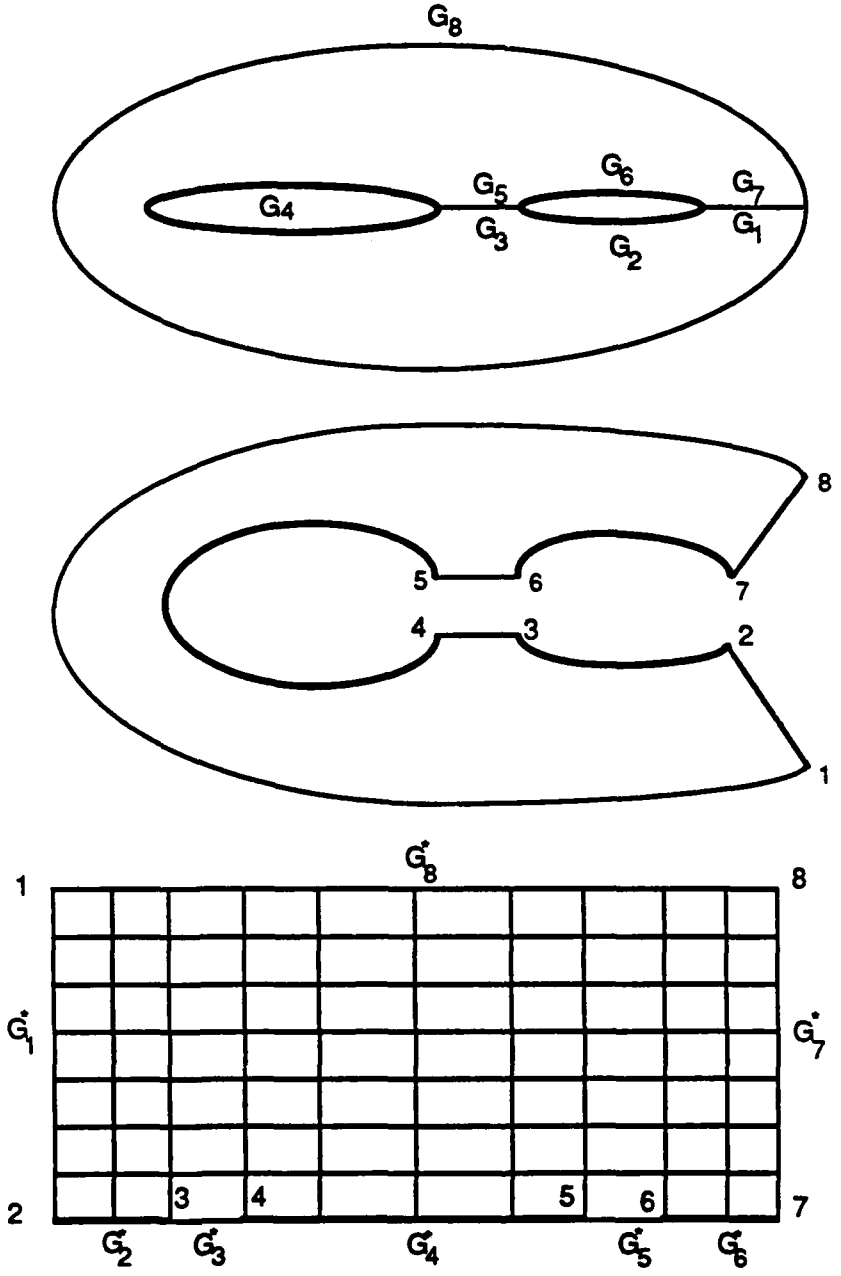


Fig. 26 An O-type grid forming for multiple bodies

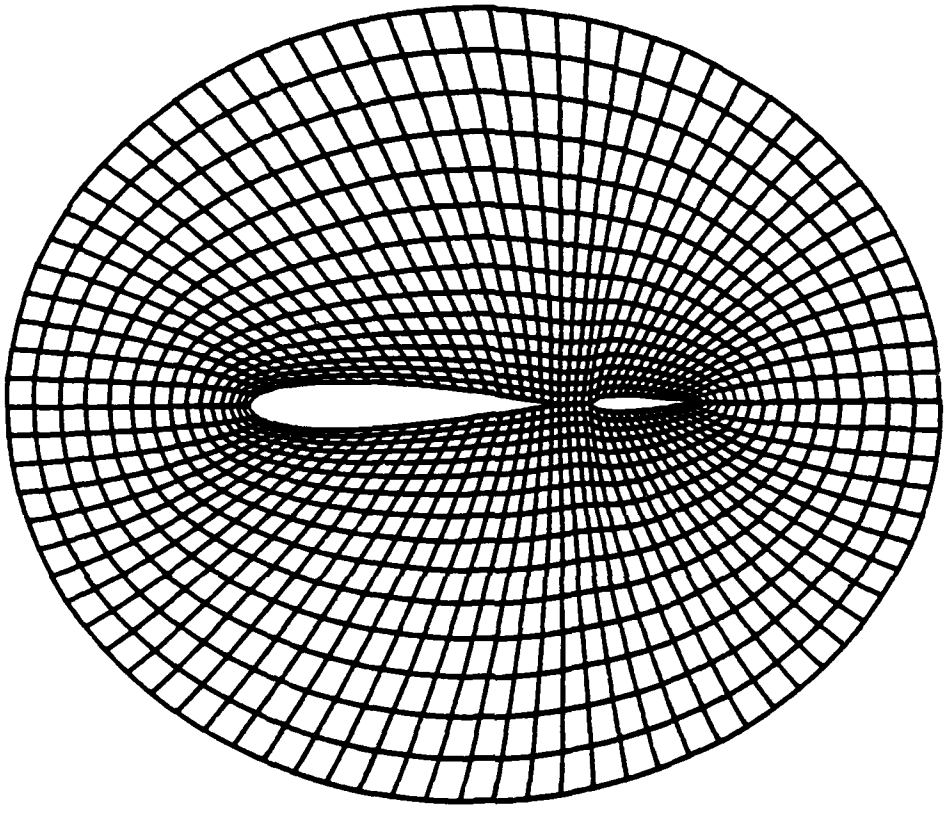


Fig.27 O-grid about two NACA 0012 airfoils generated by the Parabolic scheme (flap angle 0°)

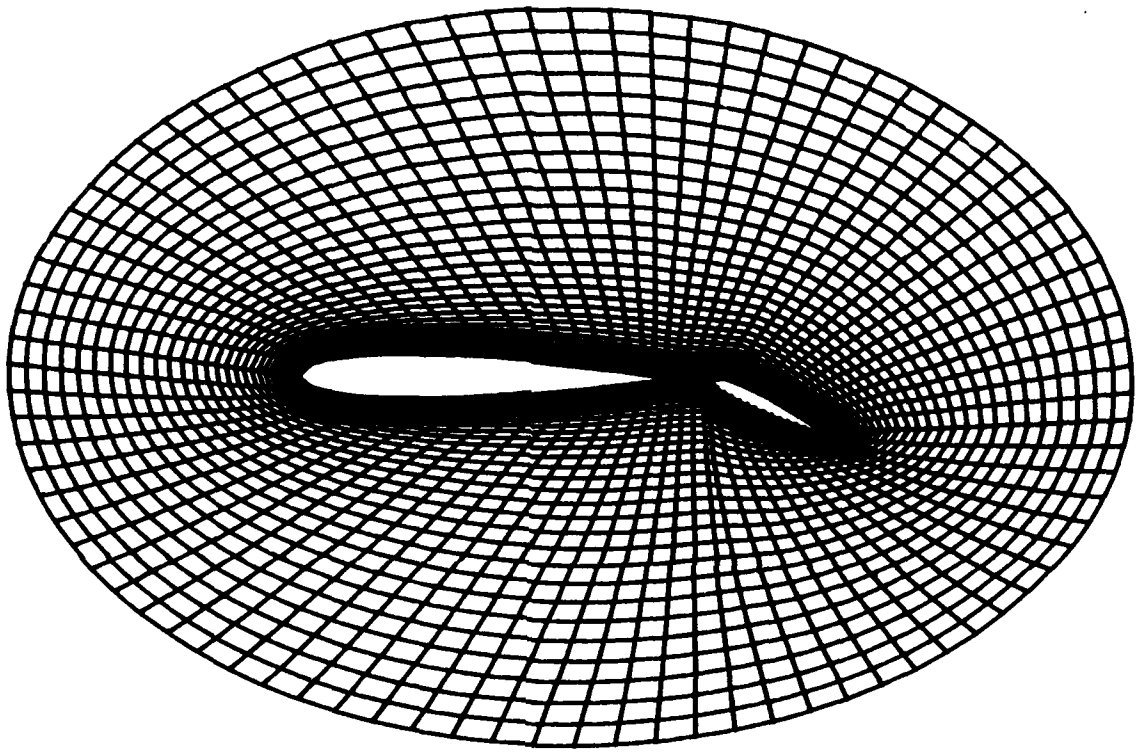


Fig. 28 O-grid about two NACA 0012 airfoils generated by the Parabolic scheme with flap angle at 25° (grid size 87 x 30, CPU time 0.623 sec.)

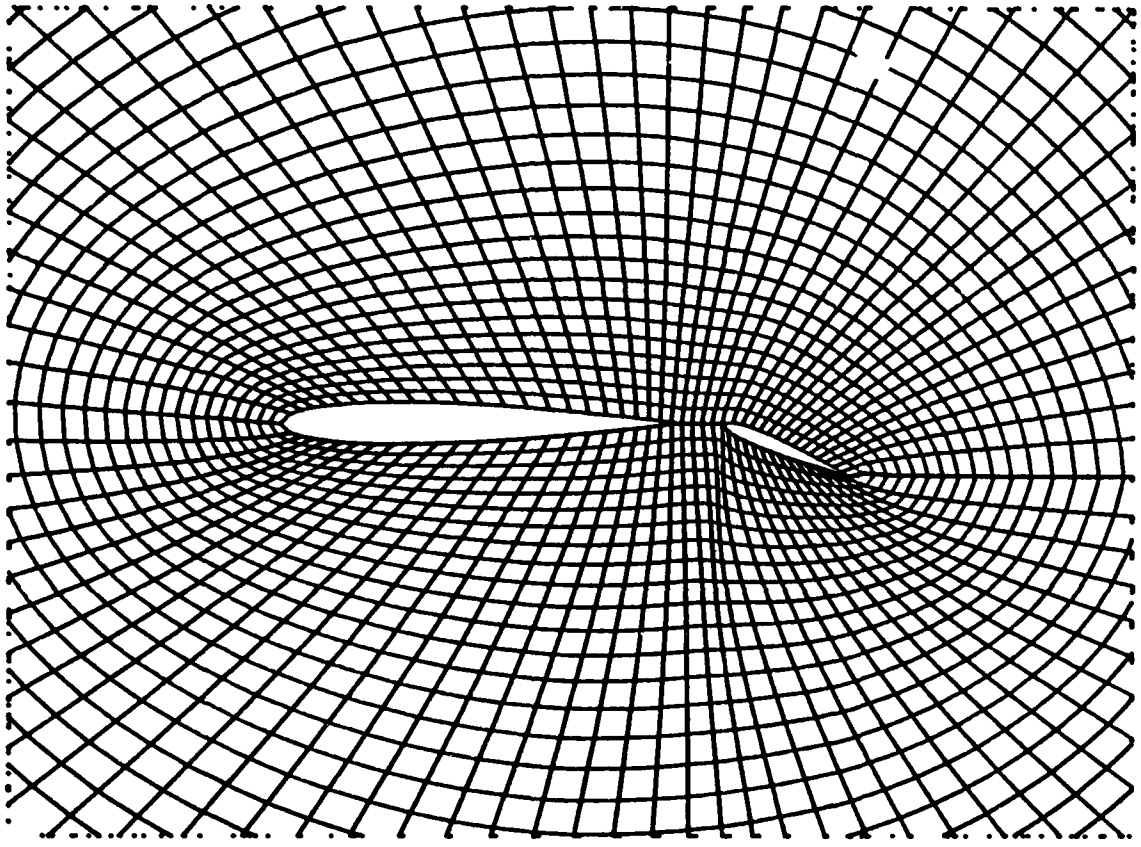


Fig. 29 O-grid about two NACA 0012 airfoils generated by the Parabolic scheme - A detail

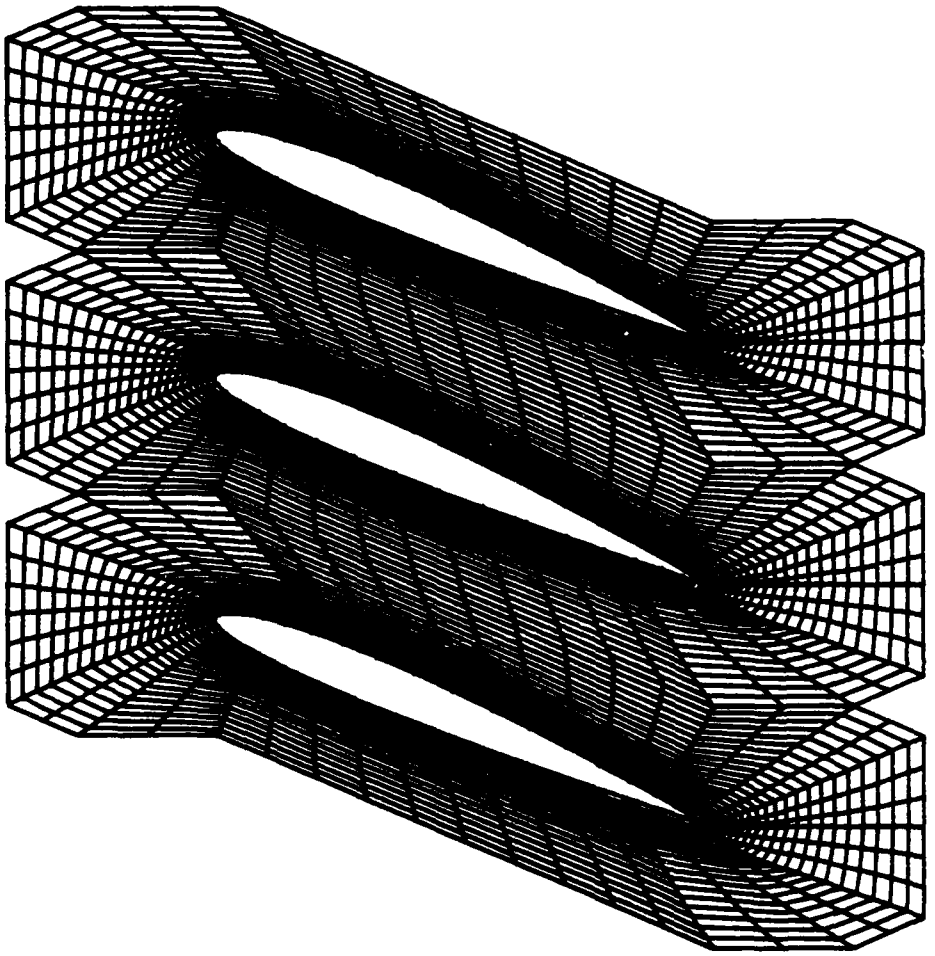


Fig.30a Cascade grid - the demand for nonsingularity at the corner points

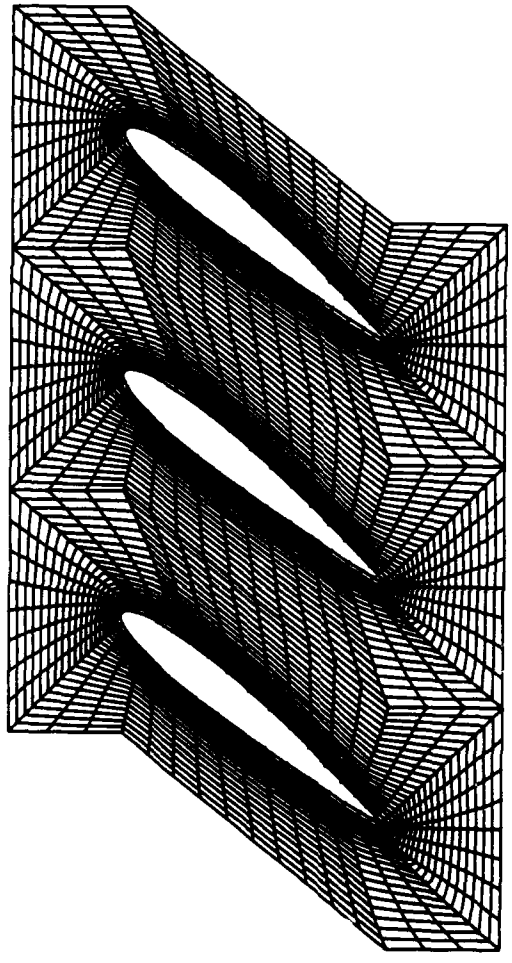


Fig.30b Cascade grid - without the demand for nonsingularity at the corner points

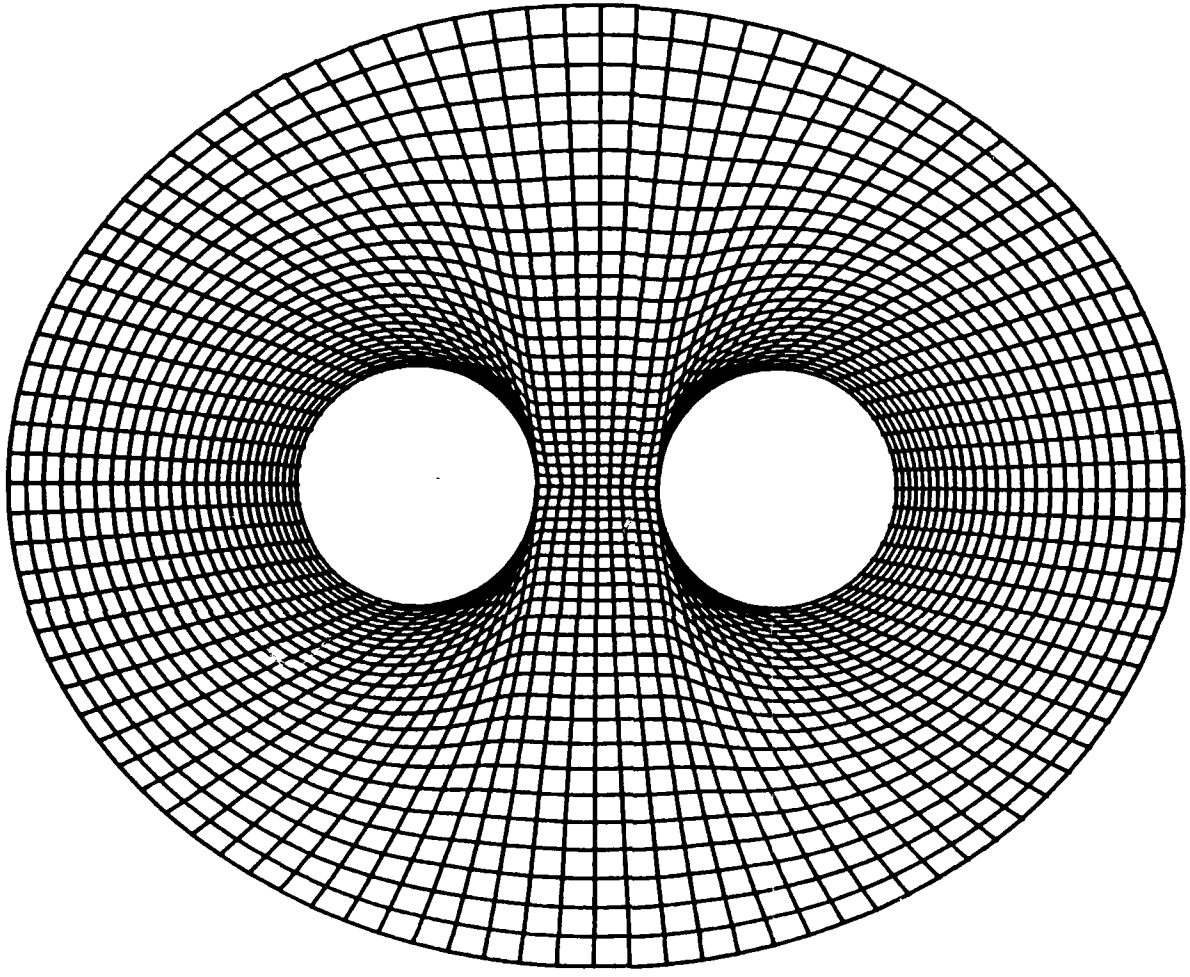


Fig. 31 O-grid about two circles generated by the Parabolic scheme

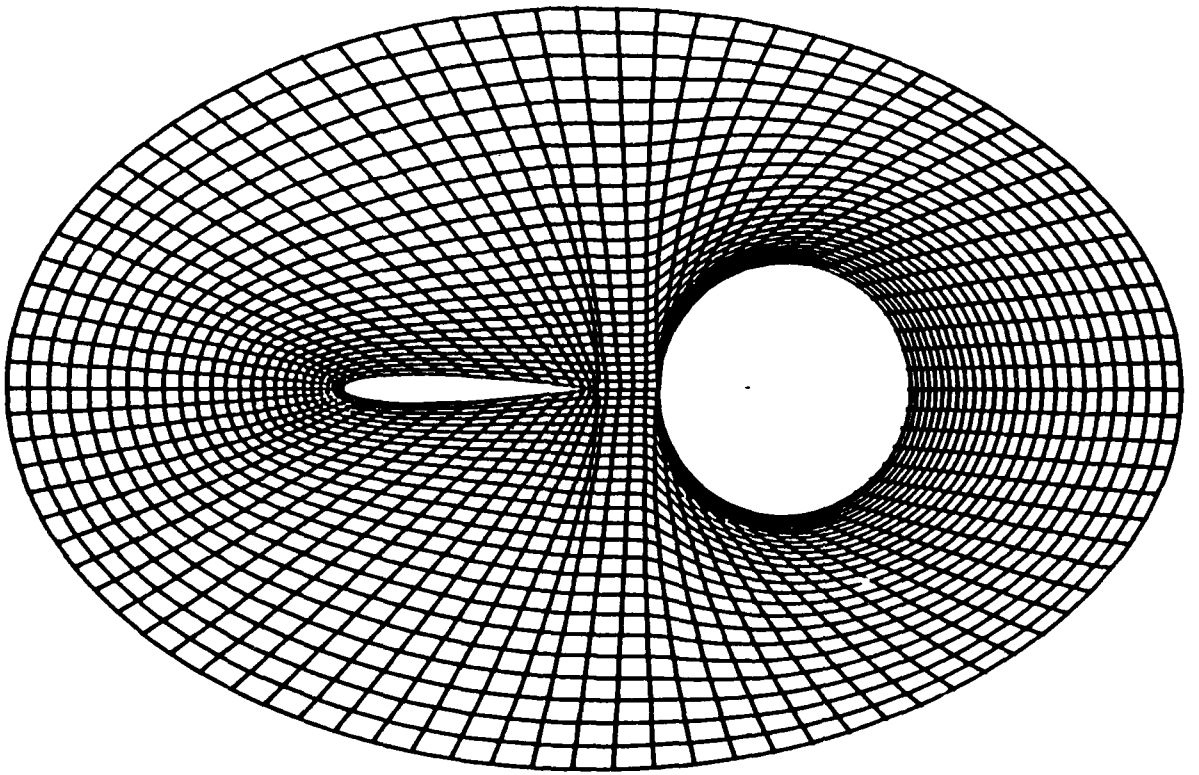


Fig. 32 O-grid about an airfol-circle configuration generated by the Parabolic scheme

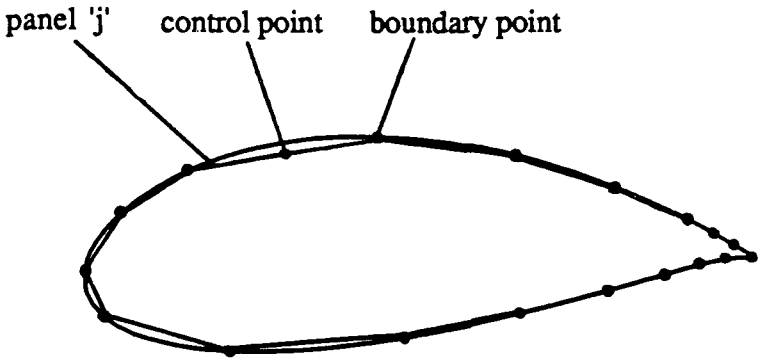


Fig. 33 Discretization of an airfoil surface

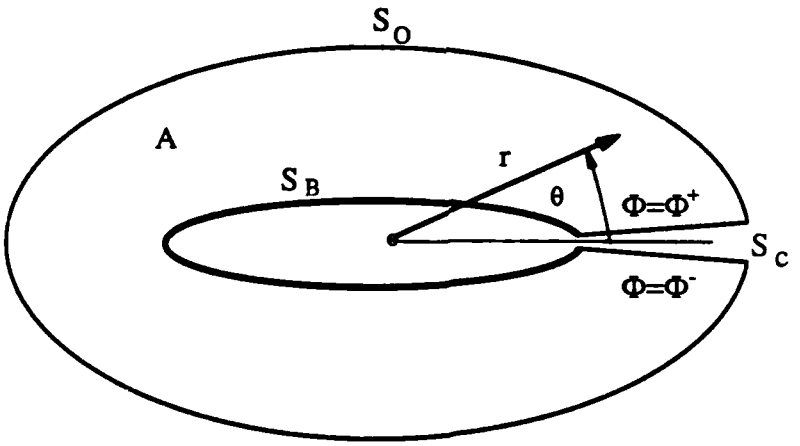


Fig. 34 Definition of integration domain

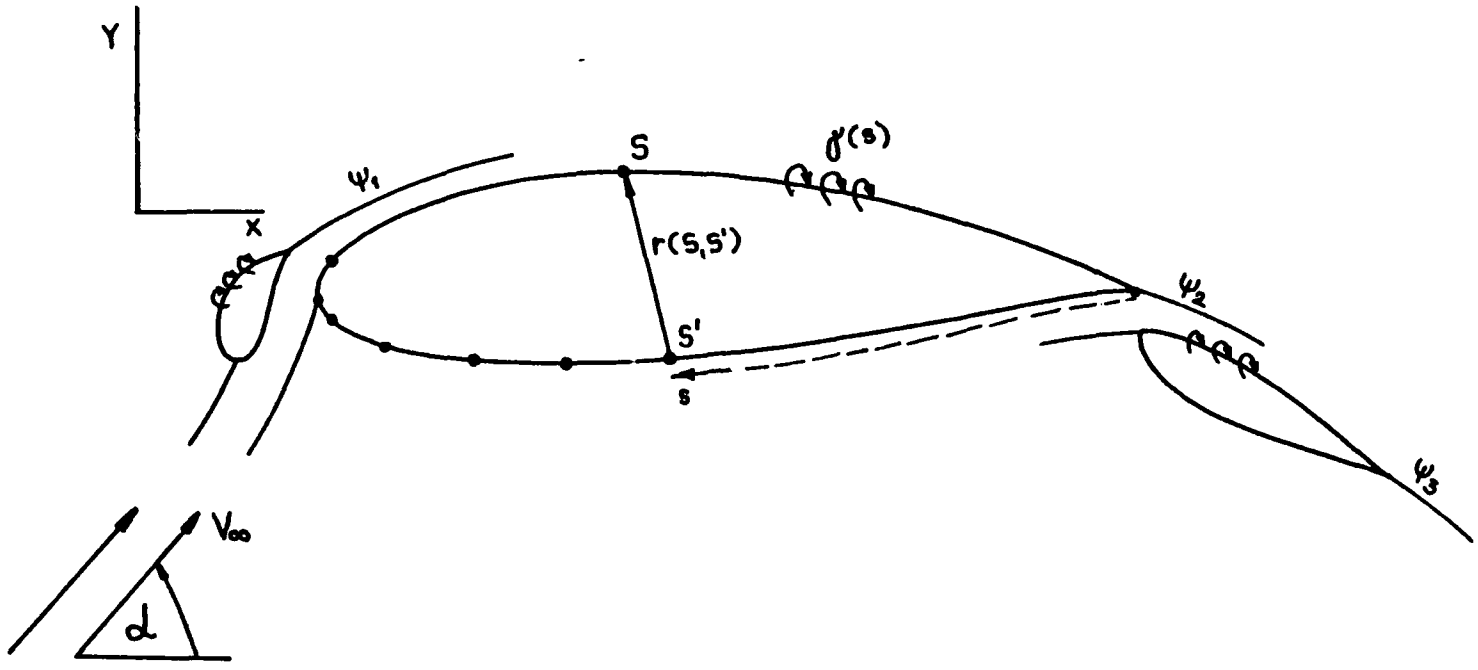


Fig. 35 Multi-component geometry and panel definition

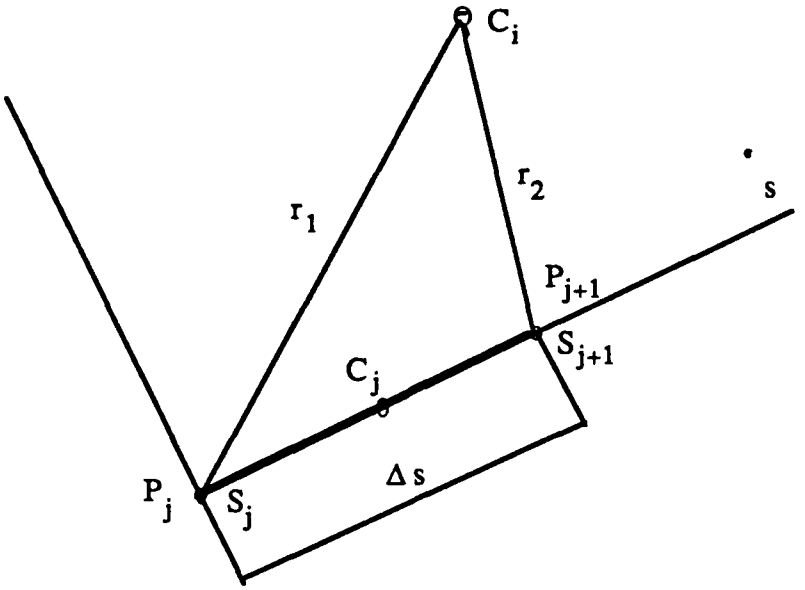


Fig. 36 Geometry definition for influence coefficients K_{ij}

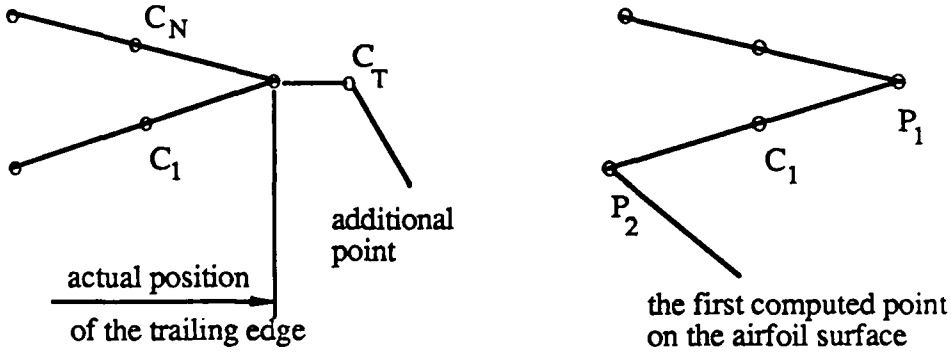


Fig. 37 a) Determination of the actual position of the trailing edge;
b) Determination of surface points

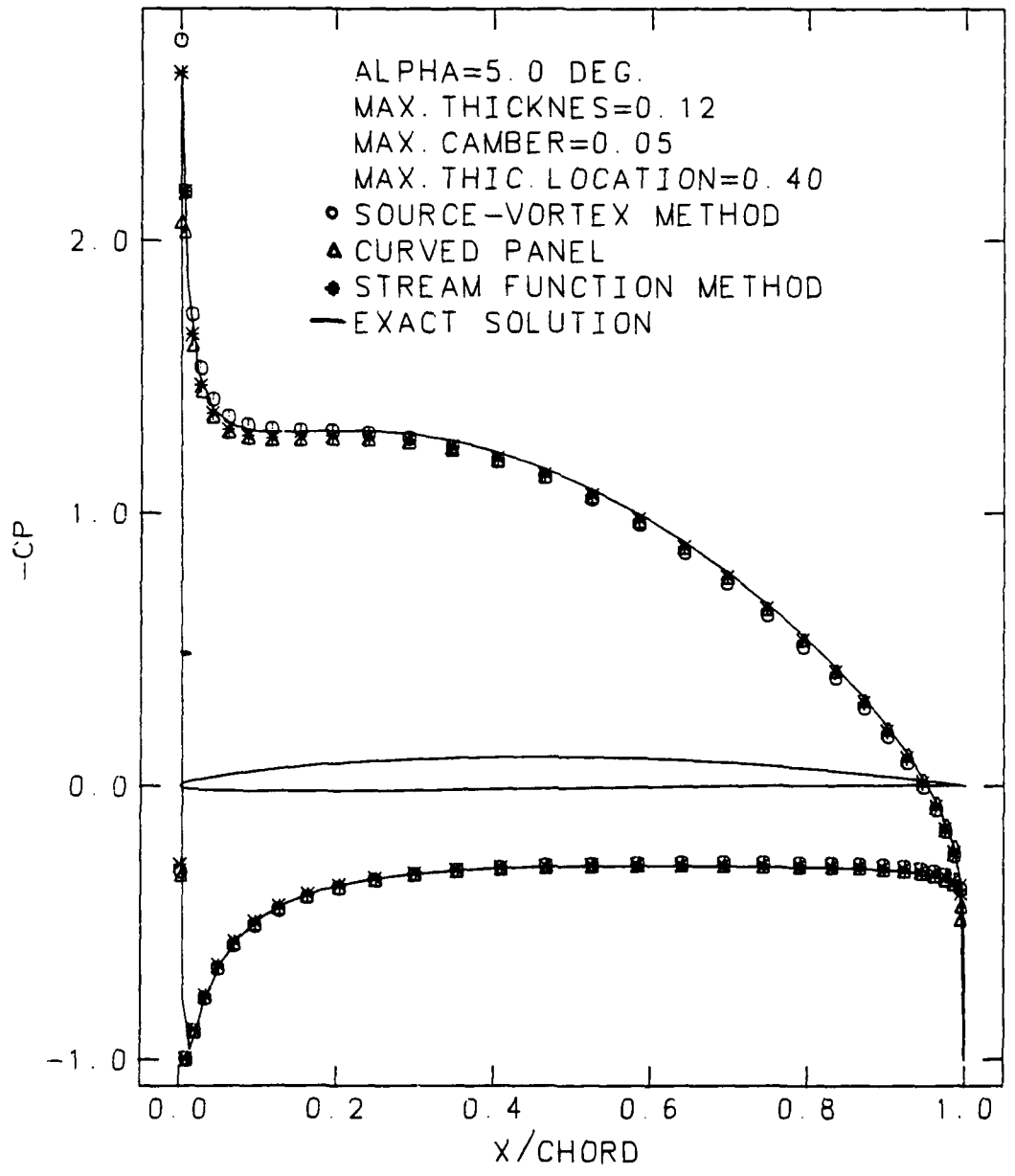


Fig. 38 Cambered Karman - Trefftz airfoil

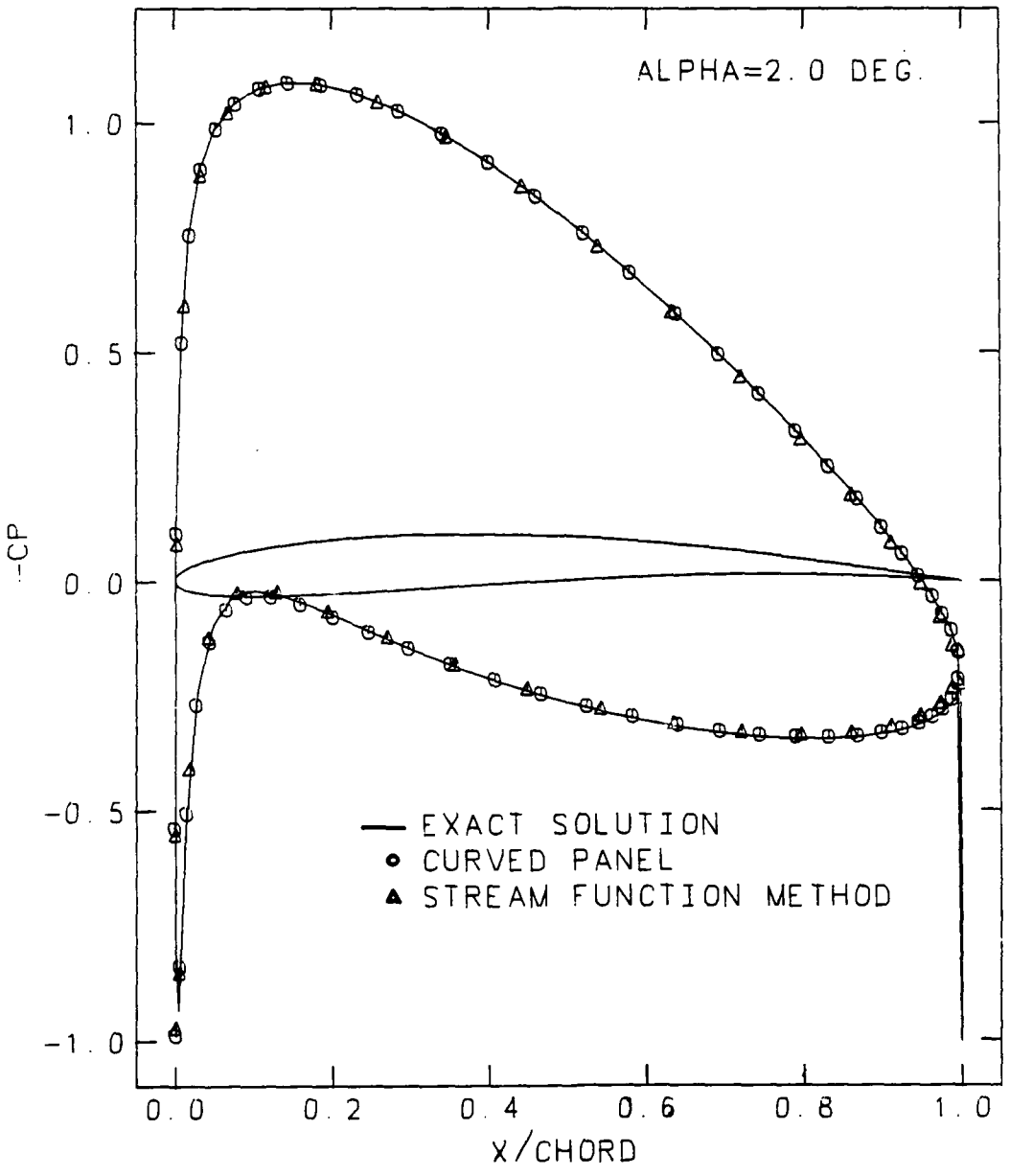


Fig. 39 Cambered Joukovski airfoil

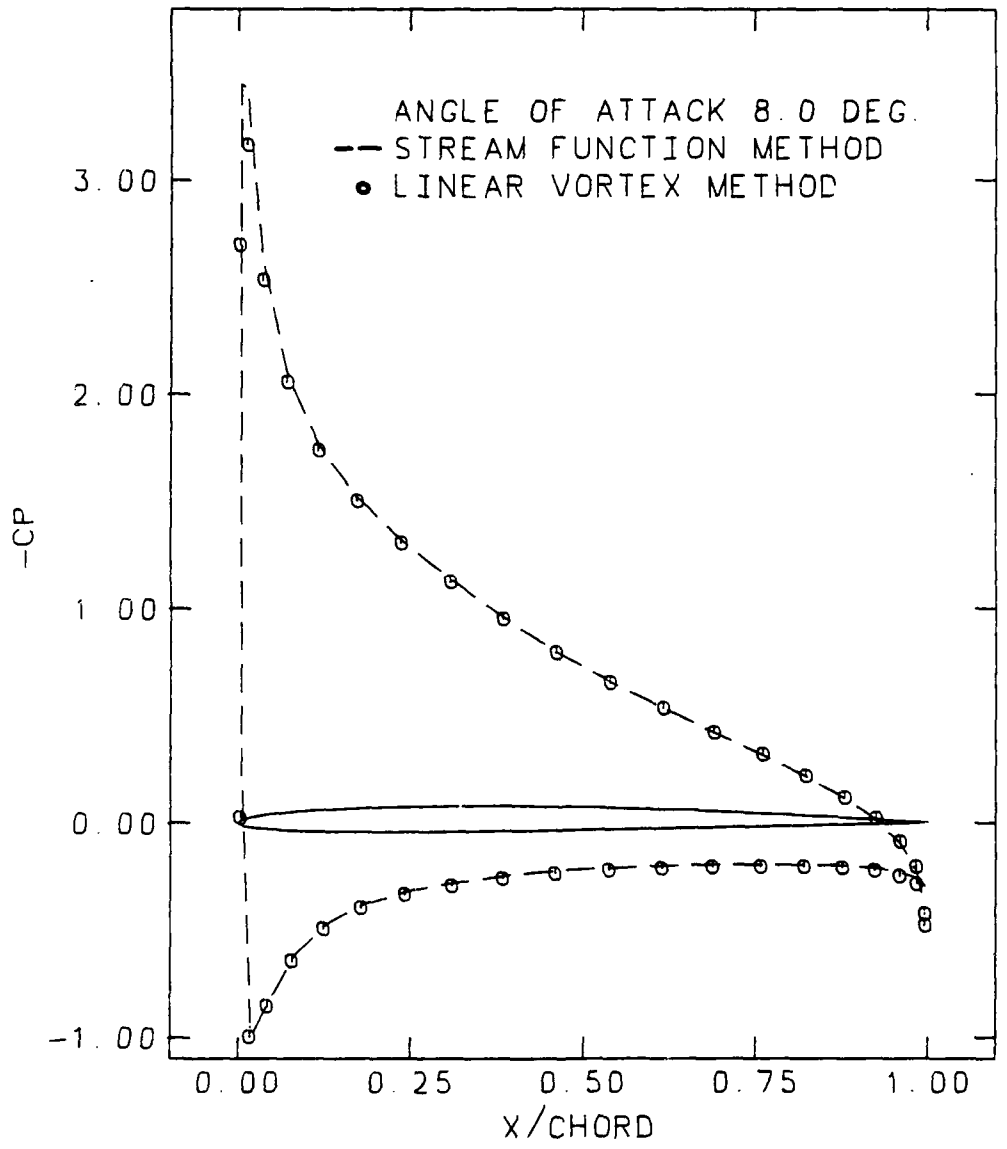


Fig. 40 NACA 2412

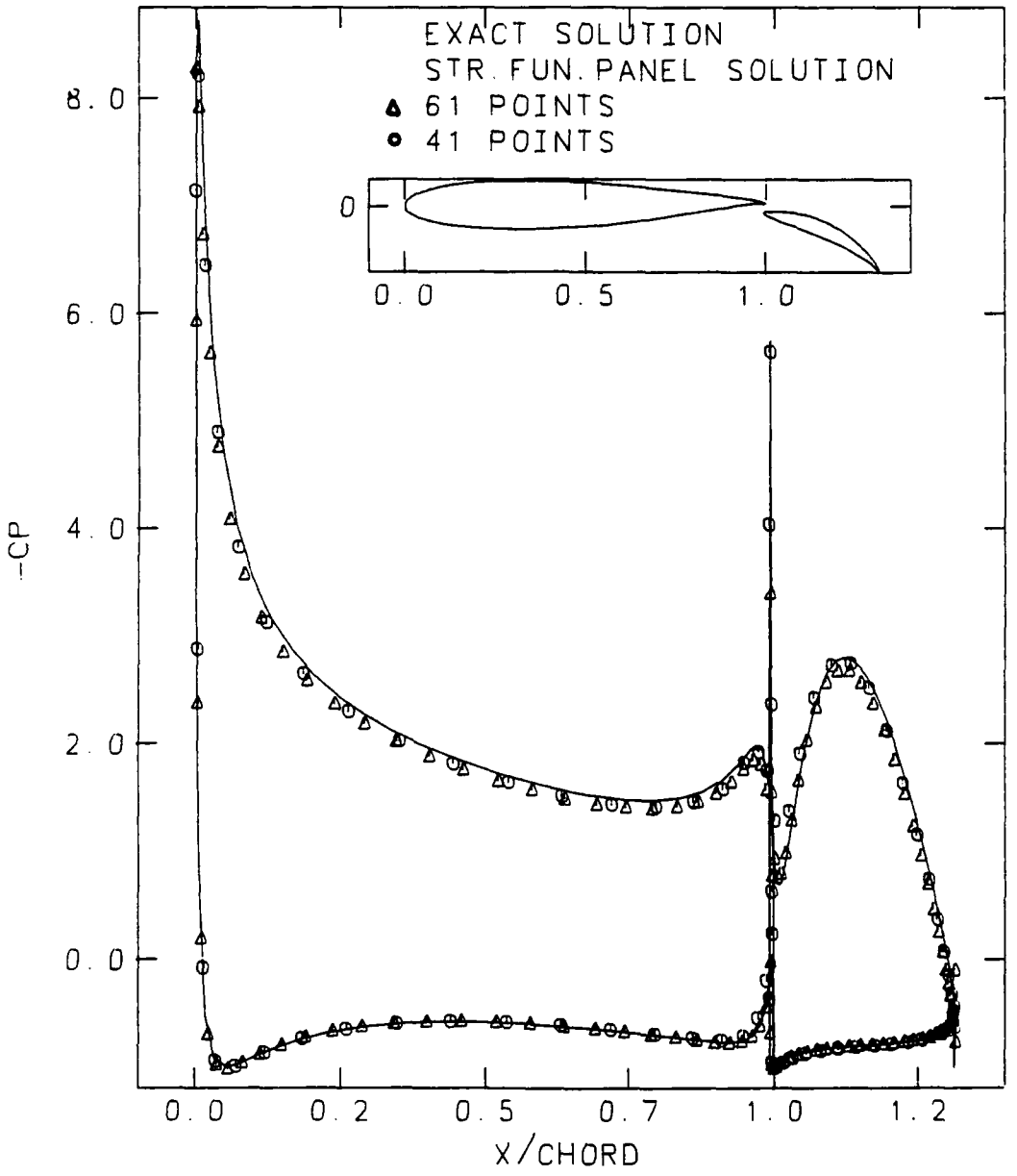


Fig. 41 William's configuration with flap angle at 30°

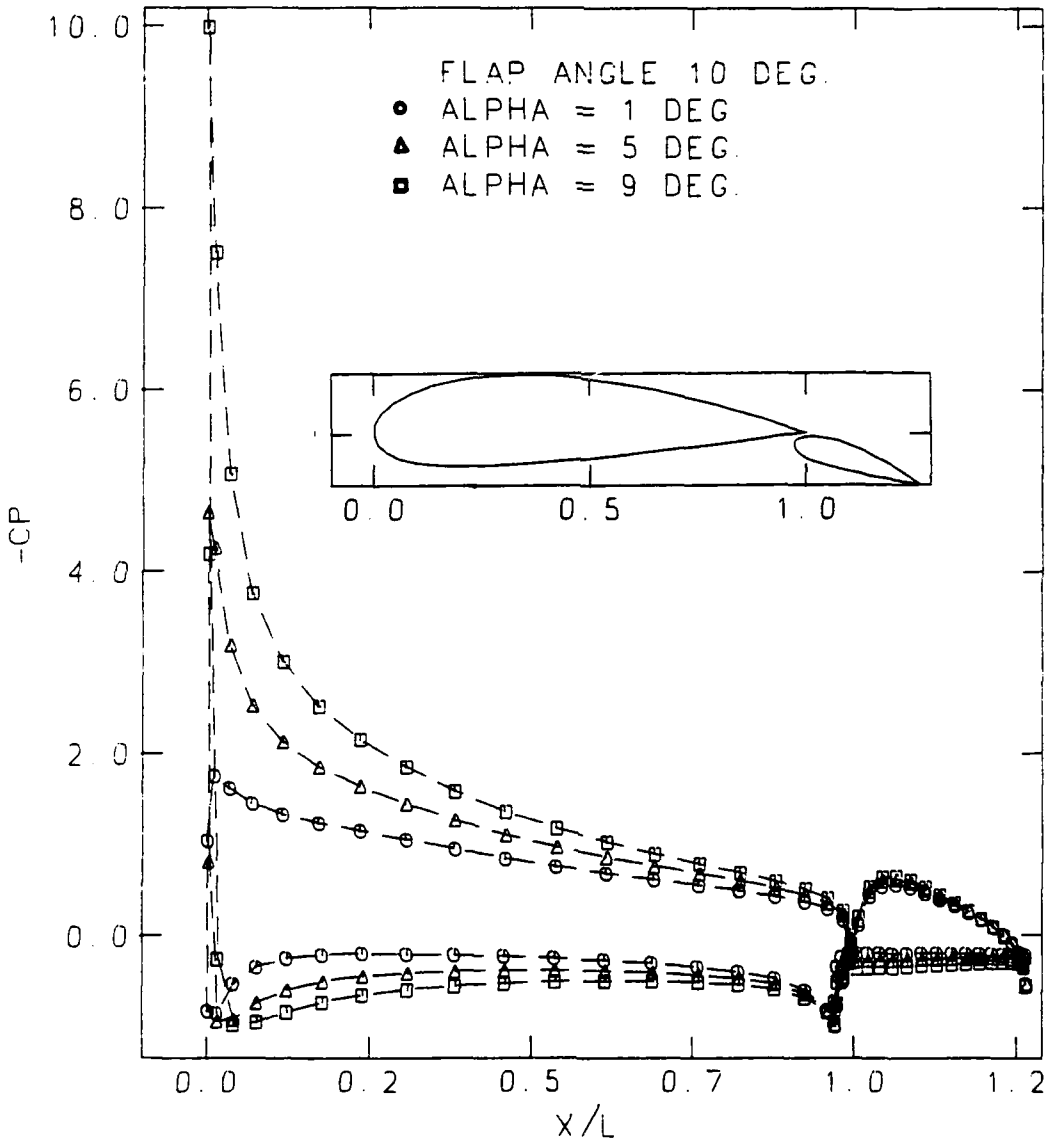


Fig. 42 Two NACA 2412 airfols at different angles of attack

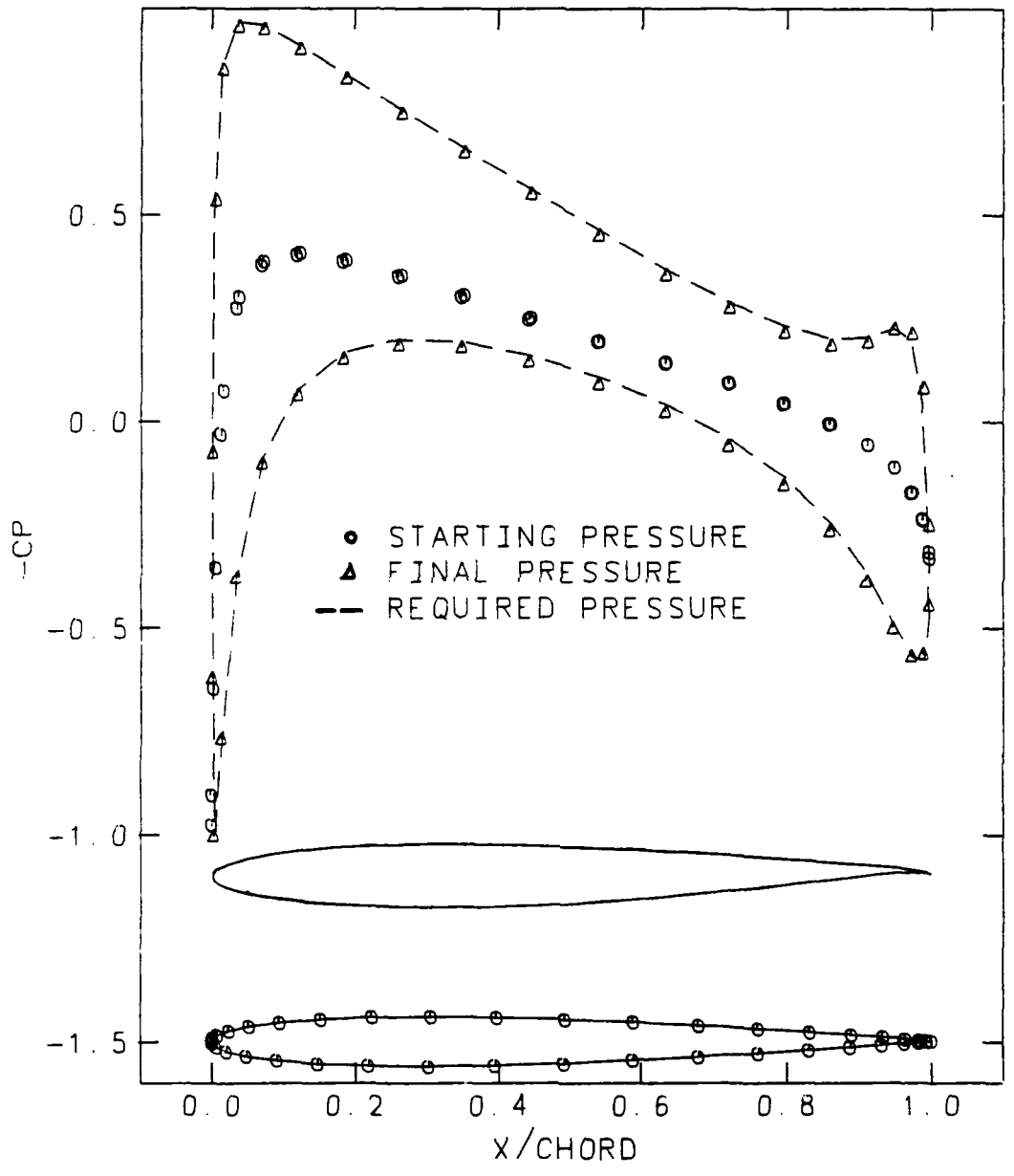


Fig. 43 Design of William's main airfoil from an NACA 0012 airfoil

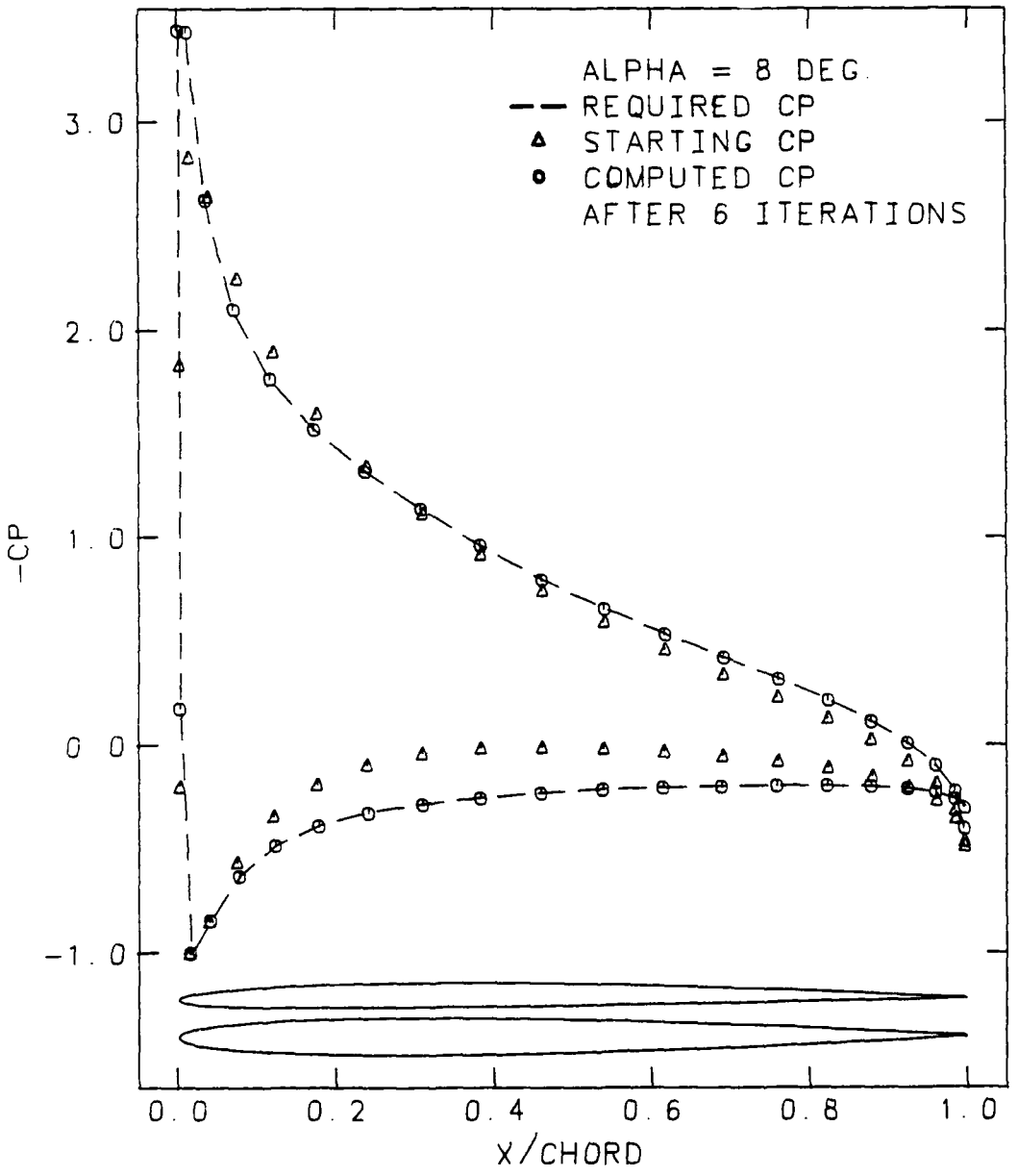


Fig. 44 Design of an NACA 2412 from an NACA 0012 airfoil

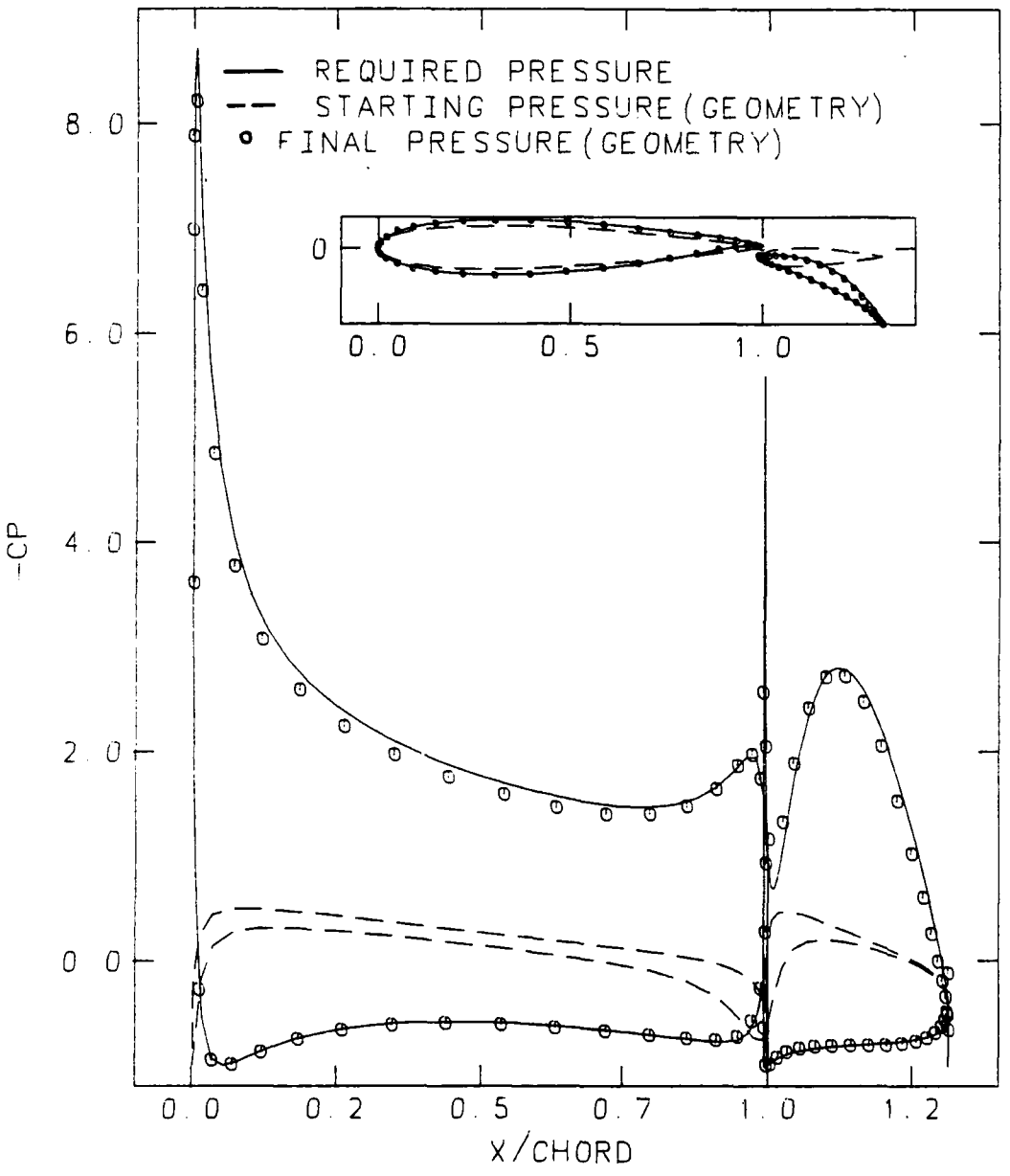


Fig. 45 Design of William's configuration from NACA 0012 and NACA 0009 airfoils

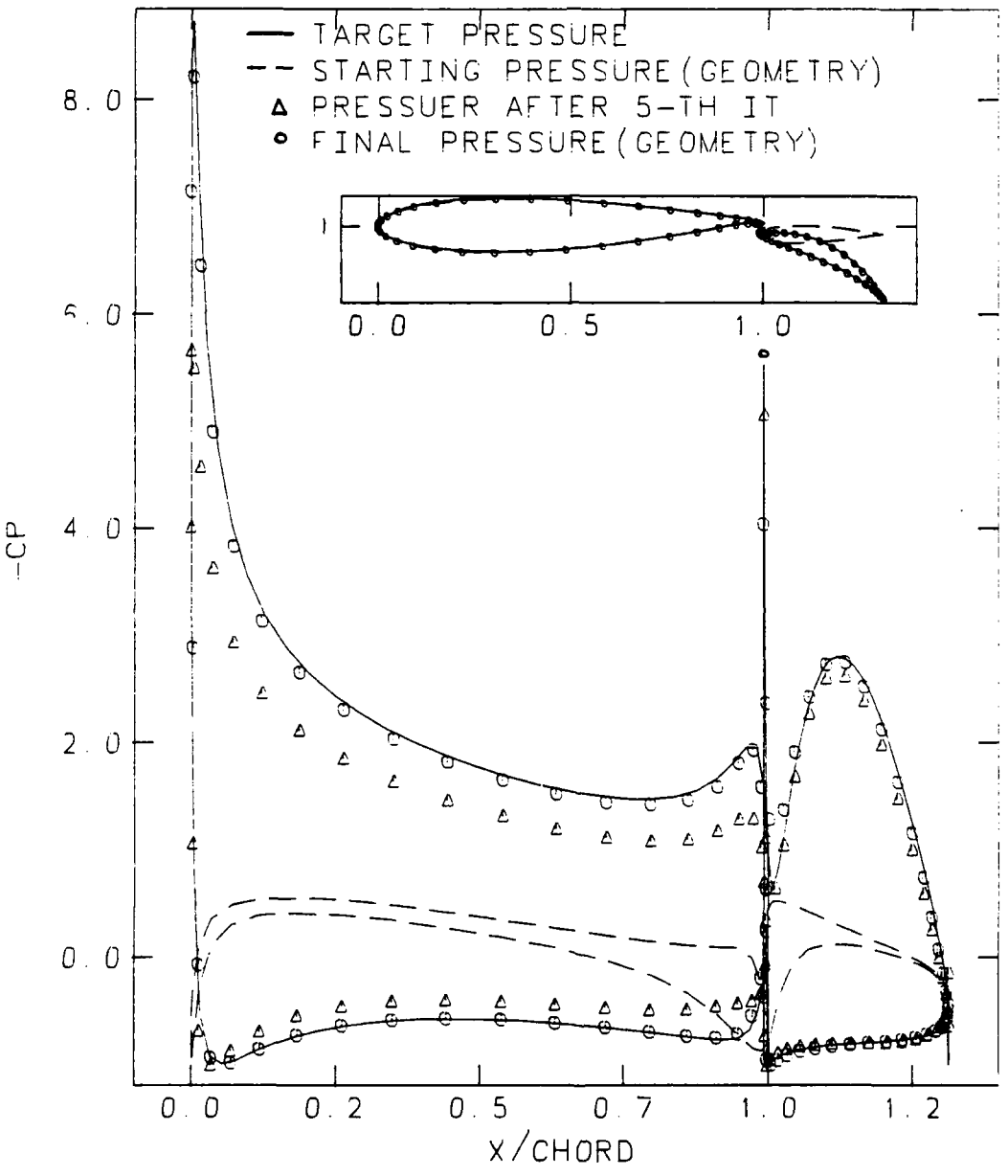


Fig. 46 Design of William's configuration from an NACA 0012 airfoil and fixed main airfoil geometry

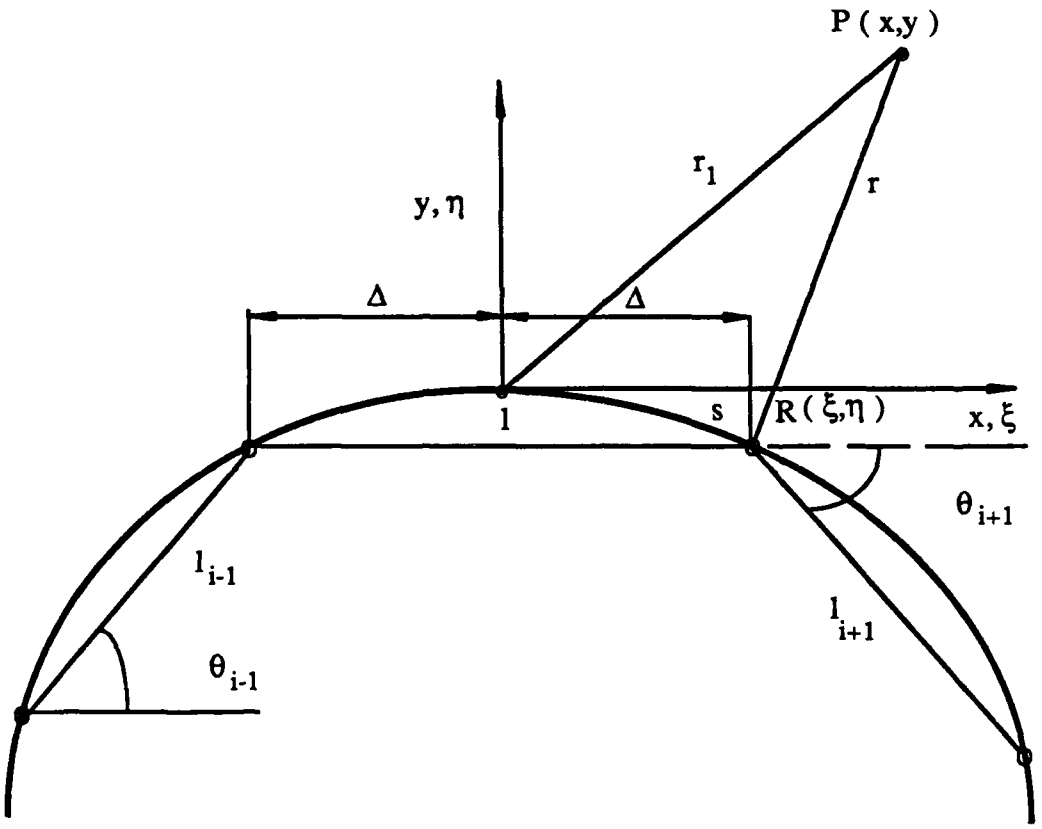


Fig. 47 Curved panel discretization

BIBLIOGRAPHY

[1] P.E. Rubbert and K,D. Lee: *"Patched Coordinate Systems"* in 'Numerical Grid Generation' edited by J.F. Thompson, North-Holland P.Co., New York 1982, pp. 235-252.

[2] F.C. Thames: *"Grid Generation of Three-Dimensional Boundary-Fitted Curvilinear Coordinate Systems for Wing/Wing-Tip Geometries Using the Elliptic Solver Method"*, in 'Numerical Grid Generation' edited by J.F. Thompson, North-Holland P.Co., New York 1982, pp. 695-716.

[3] R.E. Smith: *"Algebraic Grid Generation"*, in 'Numerical Grid Generation' edited by J.F. Thompson, North-Holland P.Co., New York 1982, pp.137-170.

[4] P.R. Eiseman: *"A Multi-Surface Method of Coordinate Generation"*, Journal of Computational Physics, vol.33, No.1, October 1979, pp. 118-150.

[5] L.E Erikson: *"Generation of Boundary-Conforming Grids Around Wing-Body Configurations Using Transfinite Interpolation"*, AIAA Journal, Vol.20, October 1982, pp.1313-1321.

[6] W.J. Gordon and L.C. Thiel: *"Transfinite Mappings and Their Application to Grid Generation"*, in 'Numerical Grid Generation' edited by J.F. Thompson, North-Holland P.Co., New York 1982., pp. 171-192.

[7] P.R. Eiseman: *"Coordinate Generation with Precise Control Over Mesh Properties"*, Journal of Computational Physics 47,1982, pp. 331-351.

[8] T.J. Akai and T.J. Mueller: *"Use of the Theodorsen Transformation to generate Orthogonal Grids for Axisymetric Bodies"*, AIAA Paper 82-0166, Orlando, Fla., Jan. 1982.

[9] N.D. Halsey: *"Potential Flow Analysis of Multielement Airfoils Using Conformal Mapping"*, AIAA Journal, Vol.17, No.12, 1979, pp.1281-1288.

[10] D. Halsey: *"Conformal grid Generation for Multicomponent Aerofoils"*, in 'Numerical Grid Generations' edited by J.F. Thompson, North-Holland P.Co., New York 1982, pp. 585-606.

[11] A. Harrington: *"Conformal Mappings onto Multiply Connected Regions with Specified Boundary Slope"*, in 'Numerical Grid Generation' edited by J.F. Thompson, North-Holland P.Co., New York 1982, pp. 602-618.

[12] D.C. Ives: *"Conformal Grid Generation"*, in 'Numerical Grid Generation' edited by J.F. Thompson, North-Holland P.Co., New York 1982, pp.107-135.

[13] D.S. Dulikravich: *"Fast Generation of Three-Dimensional Computational Boundary-Conforming Periodic Grids of C-Type"*, NACA CR-165596/83.

[14] O.L. Anderson, T.T. Davis, G.B. Hankins and D.E. Edwards: *"Solution of Viscous Internal Flows on Curvilinear Grids Generated by the Schwarz-Christoffel Transformation"*, in 'Numerical Grid Generation' edited by J.F. Thompson, North-Holland P.Co., New York 1982, pp. 507-524.

[15] L.Lapidus and G.F. Pinder: *"Numerical Solution of Partial Differential Equations in Science and Engineering"*, Princenton University Press, Princeton, NJ 1982.

[16] J.F. Thompson, F.C. Thames and C.W. Mastin: *" TOMCAT - A Code for Numerical Generation of Boundary Fitted Curvilinear Coordinate System on Fields Containing Any Number of Arbitrary Two - Dimensional Bodies"* Journal of Computational Physics 24, 1977, pp. 274-302.

[17] T.L. Holst: *"Approximate Factorization Schemes for Solving the Transonic Full Potential Equation"*, in 'Advances in Computational Transonics' edited by W.G. Habashi, Vol.4, Pineridge Press, Swansea, U.K. 1986, pp.59-82.

[18] J.L. Steger and R.L. Sorenson: *"Automatic Mesh-Point Clustering Near a Boundary in Grid Generation with Elliptic Partial Differential Equations"*, Journal of Computational Physics 33, 1979, pp. 405-410.

[19] R.L. Sorenson: *"Grid Generation by Elliptic Partial Differential Equations for Three-Element Augmentor Wing Airfoil"*, in 'Numerical Grid Generation' edited by J.F. Thompson, North-Holland P.Co.,New York 1982, pp. 653-665.

[20] R.M. Coleman: *"Generation of Boundary-Fitted Coordinate Systems Using Segmented Computational Regions"*, in 'Numerical Grid Generation' edited by J.F. Thompson, North-Holland P.Co., New York 1982, pp. 633-651.

[21] C.J. Chen and K.M. Obasih: *"Numerical Generation of Nearly Orthogonal Boundary-Fitted Coordinate System"*, in Advacements in Aerodynamics, Fluid Mechanics and Hydraulics' edited by R.E.A. Arndt, H.G. Stefan, C. Farell and S.M. Peterson, Amercan Society of Civil Engineers, New York 1986, pp.586-592.

[22] M. Visbal and D. Knight: *"Generation of Orthogonal and Nearly Orthogonal Coordinates with Grid Control Near Boundaries"*, AIAA Journal, Vol. 20, No. 3, March 1982, pp. 305-306.

[23] C.F. Shieh: *"Three-Dimensional Grid Generation Using Poisson Equation"*, in 'Numerical Grid Generation' edited by J.F. Thompson, North-Holland P.Co., New York 1982, pp.687-694.

[24] J.F. Thompson, Z.U.A. Warsi and C.W.Mastin: *"Numerical Grid Generation - Foundations and Applications"*, North - Holland, New York 1985.

[25] J.L. Steger and R.L. Sorenson: *"Use of Hyperbolic Partial Differential Equations to Generate Body Fitted Coordinates"*, in 'Numerical Grid Generation Techniques' NASA CP-2166, 1980, pp.463-478.

[26] J.L. Steger and D.S. Chaussee: *"Generation of Body Fitted Coordinates Using Hyperbolic Partial Differential Equations"*, SIAM Journal of Scientific and Statistical Computing, Vol.1, 1980, pp. 431-437.

[27] P. Kutler: *"A Perspective of Theoretical and Applied Computational Fluid Dynamics"*, AIAA Journal, Vol.23, No.3, 1985, pp. 328-341.

[28] S. Nakamura: *"Marching Grid Generation Using Parabolic Partial Differential Equations"*, in 'Numerical Grid Generation' edited by J.F.Thompson, North-Holland P. Co.,New York 1982, pp. 775-783.

[29] S. Nakamura: *"Noniterative Grid Generation Using Parabolic Difference Equations for Fuselage-Wing Flow Calculations"*, Lecture Notes in Physics, Vol.170, Springer-Verlag, 1982, pp.396-401.

[30] T.A. Edwards: *"Noniterative Three Dimensional Grid Generation Using Parabolic Partial Differential Equations"*, AIAA-85-0485.

[31] J.U. Brackbill: *"'Coordinate System Control: Adaptive Meshes"*, in 'Numerical Grid Generation' edited by J.F. Thompson, North-Holland P.Co., New York 1982, pp. 277-294.

[32] J. Saltzman and J. Brackbill: *"Application and Generalization of Variational Methods for Generating Adaptive Meshes"* in 'Numerical Grid Generation' edited by J.F. Thompson, North-Holland P.Co., New York 1982, pp. 865-883.

[33] D.A. Anderson and M.M. Rai: *"The Use of Solution Adaptive Grids in Solving Partial Differential Equations"*, in 'Numerical Grid Generation' edited by J.F. Thompson, North-Holland P.Co., New York 1982, pp. 317-338.

[34] G.M. Johnson: *"Accelerated Solution of the Steady Euler Equations', in 'Advances in Computational Transonics"* edited by W.G. Habashi, Vol.4, Pinderidge Press, Swansea, U.K. 1986, pp. 473-501.

[35] J.L. Hess and A.M.O. Smith: *"Calculation of Potential Flow About Arbitrary Bodies"*, Progress in Aeronautical Sciences' Vol.8, Pergamon Press 1967.

[36] J.L. Hess: *"Higher Order Numerical Solution of the Integral Equation for the Two-Dimensional Neuman Problem"*, Computer Methods in Applied Mechanics and Engineering 2, 1973, pp. 1-15.

[37] J.L.Hess: *"Improved Solution for Potential Flow About Arbitrary Axisymetric Bodies by the Use Of a Higher Order Surface Singularity Method"*, Computer Methods in Applied Mechanics and Engineering 5, 1975, pp. 297-308.

[38] J.L. Hess: *"The Use of Higher-Order Surface Singularity Distributions to Obtain Improved Potential Flow Solutions for Two-Dimensional Lifting Airfoils"*, Computer Methods in Applied Mechanics and Engineering 5, 1975, pp. 1-35.

[39] B. Maskew: *"Prediction of Subsonic Aerodynamics Characteristics: A Case of Low Order Panel Methods"*, J. Aircraft , Vol. 19, No.2, 1982.

[40] D.R. Bristow and G.G. Grose: *"Development of Panel Methods for Subsonic Analysis and Design"*, NASA CR-3234,1980.

[41] A.I. Ormsbee and A.W. Chen: *"Multiple Element Airfoils Optimized for Maximum Lift Coefficient"*, AIAA Journal, Vol.10,No.12, 1972, pp. 1620-1624.

[42] J.L. Kennedy and D.J. Marsden: *"A Potential Flow Design Method for Multicomponent Airfoil Sections"*, J. Aircraft, Vol.15, No.1, 1978, pp.47-53.

[43] H.N.V. Dutt and A.K. Sreekauth: *"Design of Airfoils for Prescribed Pressure Distribution in Viscous Incompressible Flows"*, Aeronautical Quartely, Vol. XXXI , Part I, 1980, pp. 42-55.

[44] E. Greff and J. Mantel: *"An Engineering Approach to Inverse Transonic Wing Design Problem"*, Communications in Applied Numerical Methods, Vol.2, 1986, pp.47-56.

[45] C.A. Brebbia: *"Boundary Element Techniques in Engineering"* Pentech Press, London 1978.

[46] G.F. Carey and S.W. Kim: *"Lifting Aerofoil Calculation Using the Boundary Element Method"*, International Journal for Numerical Methods in Fluids, Vol. 3, 1983, pp.481-492.

[47] A. Saddhoo and I.M. Hall: *"Test Cases for Plane Potential Flow Past Multi-Element Airfoils"*, The Aeronautical Journal, Vol. 89,No.890, 1985, pp. 403-414.

[48] J. Moran: *"An Introduction to Theoretical and Computational Aerodynamics"*, John Wiley and Sons, 1984.

[49] J.F. Thompson, F.C. Thames and C.W. Mastin: *"Boundary-Fitted Curvilinear Coordinate Systems for Solution of Partial Differential Equations on Fields Containing any Arbitrary Two-Dimensional Bodies"*, NACA CR 2729/77.

[50] F.C. Thames, J.F. Thompson, C.W. Mastin and R.L. Walker: *"Numerical Solution for Viscous and Potential Flaw About arbitrary Two-Dimensional Bodies Using Body-Fitted Coordinate System"*, Journal of Computational Physics 24, pp. 245-273, 1977.

[51] D.A. Anderson, J.C. Tannehill, R.H. Pletcher: *"Computational Fluid Mechanics and Heat Transfer"*, Hemisphere P.Co., 1984.

[52] J.F. Thompson: *"Grid Generation Techniques in Computational Fluid Dynamics"*, AIAA Journal ,Vol.22, No.11, 1984, pp.1505-1523.

[53] J.F. Thompson: *"Elliptic Grid Generation"*, in 'Numerical Grid Generation' edited by J.F. Thompson, North-Holland P.Co., New York 1982, pp.79-105.

[54] M.B. Aston and J.W. Thomas: *"An Implicit Scheme for Water Wave Problems"*, in 'Numerical Grid Generation' edited by J.F. Thompson, North-Holland P.Co., New York 1982, pp. 809-818.

[55] K.D. Klevenhusen: *"2-D Elliptic Grid Generation Using a Singularity Method and its Application to Transonic Interference Flows"*, in 'Numerical Grid Generation' edited by J.F. Thompson, North-Holland P.Co., New York 1982, pp.739-760.

[56] D.C. Ives: *"A Modern Look at Conformal Mapping, Including Multiply Connected Regions"*, AIAA Journal , Vol.14, 1976, pp. 1006-1011.

[57] R.L. Sorenson and J.L Steger: *"Numerical Grid Generation for Two-Dimensional Grids by Poisson Equations with Orthogonality Control at Boundaries"*, Numerical Grid Generation Techniques, NASA CP-2166, 1980, pp. 449-460.

[58] J.F. Thompson, Z.U.A. Warsi and C.W. Mastin: *"Boundary-Fitted Coordinate Systems for Numerical Solution of Partial Differential Equations - A Review"*, Journal of Computational Physics 47, 1982, pp.1-108.

[59] G.F. Carey and J.T Oden: *"Finite Elements: A Second Course"*, Vol. 2, Prentice-Hall, New Jersey 1983.

[60] J. Moran, V. Cole and D. Wahl: *"Analysis of Two-Dimensional Incompressible Flows by Subsurface Panel Method"*, AIAA Journal, Vol.18, No.6, 1980, pp.526-533.

[61] T.D. Beaty and J.C. Narramore: *"Inverse Method for the Design of Multielement High-Lift Systems"*, J. Aircraft Vol.13, No.6, 1976, pp.393-398.

[62] T. Seebohm and B.G. Newman: *"A Numerical Method for Calculating Viscous Flow Round Multiple-Section Airfoils"*, Aeronautical Quartely ,Vol. XXVI, part 3, 1975, pp. 176-188.

[63] J.L. Kennedy and D.J. Marsden: *"Potential Flow Velocity Distributions on Multi - Component Airfoil Sections"*, Canadian Aeronautics and Space Journal', Vol.2, 1976, pp. 243-256.

[64] T. Fujinami: *MS Thesis*, Department of ASE and EM, University of Texas at Austin, 1986.

[65] A.M. Kuethe and C.Y. Chow: *"Foundations of Aerodynamics"*, John Wiley and Sons, 1986.

[66] J.B. Malone: *"A Subsonic Panel Method for Iterative Design of Complex Aircraft Configurations"*, J. Aircraft, Vol.19, No.10, 1982, pp. 820-825.

VITA

Mato F. Siladic [redacted] the son of Slavica Siladic and Franjo Siladic. He attended high school at Sarajevo and undergraduate studies at University Beograd, Beograd, where he received a Bachelor of Science degree in Aerospace Engineering in October 1982. During the following years he served as a member of The Yugoslav National Army.

The author entered graduate school at The University of Texas at Austin in September 1985.

[redacted]